EDUFACESYNC: AI-based Expression Detection System

Session: 2022-2026

Submitted by:

Muskan Altaf 2022-CS-116
Rida Batool 2022-CS-118

Submitted to:

Sir Samyan Wahla

Course:

Artificial Intelligence

Department of Computer Science

University of Engineering and Technology

Lahore, Pakistan

1 Executive Summary

EduFaceSync is an AI-based facial expression detection system integrated with an LMS (Learning Management System). The system has been developed using Python for model training, HTML, CSS, and JavaScript for the frontend, and MySQL for backend management. This system is expected to enhance the teaching strategy by giving real-time insights into students' engagement and emotional state while teaching through online classes. This report describes the purpose, scope, features, and modules explained in the SRS, outlining how these will contribute to better teaching methods.

Contents

1 Executive Summary **2**

2 Introduction **7**
2.1 Purpose . . . 7
2.2 Scope . . . 7
2.3 Benefits and Goals . . . 7

3 System Modules and Their Role in Improving Teaching Strategies **7**
3.1 User Management Module . . . 7
3.1.1 Registration and Authentication . . . 7
3.1.2 Role Assignment . . . 8
3.2 Course Management Module . . . 8
3.2.1 Course Creation and Management (Teacher) . . . 8
3.2.2 3.2.2 Enrollment (Student) . . . 8
3.3 Live Class Facilitation Module . . . 8
3.3.1 Virtual Classes . . . 8
3.3.2 Expression Detection . . . 9
3.4 Reporting Module . . . 9
3.4.1 Attendance Reports . . . 9
3.4.2 Emotion Detection Reports . . . 9

4 Model Training and Evaluation **10**
4.1 Data Augmentation and Training Process . . . 10
4.2 Model Architecture . . . 10
4.3 Performance Evaluation . . . 10
4.4 Classification Report and Model Evaluation Metrics . . . 10
4.5 Explanation of Evaluation Metrics . . . 11

5 Wireframes **11**
5.1 Landing Page . . . 12
5.2 Sign-Up Page . . . 12
5.3 Login Page . . . 13
5.4 Reset Password . . . 13
5.5 Teacher Dashboard . . . 14
5.6 Manage Course Page . . . 14
5.7 Create Live Class Page . . . 14
5.8 Attendence Report Page . . . 15
5.9 Expression Report Page . . . 16
5.10 Student Dashboard Page . . . 16
5.11 View Available Course Page . . . 17
5.12 Join Live class Page . . . 17

5.13 View Attendance Check Page 18

6 Functional Requirements of EduFaceSync System 18
6.1 User Sign-Up 18
6.2 User Login 18
6.3 Course Creation (Teacher) 19
6.4 Course Enrollment (Student) 19
6.5 Live Class Creation (Teacher) 20
6.6 Student Joins Live Class 20
6.7 Automatic Attendance Tracking 20
6.8 Emotion Detection Report Generation 21

7 Non-Functional Requirements 21

8 Emotion Detection Model Using Convolutional Neural Networks (CNN) 22
8.1 Model Architecture 22
8.2 Data Augmentation 23
8.3 Model Training and Evaluation 23
8.4 Model Usage for Real-Time Emotion Detection 24

9 Conclusion 24

List of Figures

1 Main Page 12
2 Sign-up page 12
3 Login-page 13
4 Reset password 13
5 Teacher dashboard 14
6 Add,Update, Delete existing course 14
7 Create live class 15
8 View Attendance report 15
9 Expression Report 16
10 Student Dashboard 16
11 Student can view All available Courses 17
12 Join Live class 17
13 View Attendance 18

List of Tables

1 Model Classification Report 11

2 Introduction

2.1 Purpose

The EduFaceSync project empowers teachers to monitor and analyze students' engagement based on facial expressions detected. Real-time feedback allows the teacher to adjust the teaching strategies effectively to achieve an interactive and engaging online learning environment.

2.2 Scope

EduFaceSync provides:

- Facial expression detection during real-time online classes.
- Detailed dashboards for teachers and students.
- Course enrollment and management for students and teachers.
- Attendance and engagement reports.

2.3 Benefits and Goals

- Increasing the participation and engagement of students.
- Giving teachers actionable insights to make appropriate teaching strategies.
- Offering detailed post-class analytics for performance evaluation.

3 System Modules and Their Role in Improving Teaching Strategies

3.1 User Management Module

This module ensures efficient user authentication and role-based access control.

3.1.1 Registration and Authentication

Description: Teachers and students can register and log in to the system securely. Python handles input validation, while MySQL stores user credentials.

How it Enhances Teaching Strategies:

- Ensures only authorized users access the system, thus maintaining data integrity.
- Role-specific access allows teachers to focus on analytics and course management while students focus on their learning experience.

3.1.2 Role Assignment

Description: Students and teachers are assigned different roles, which are granted specific permissions.

How it Improves Teaching Methods:

- Teachers are allowed access to student engagement data, which enables them to make targeted teaching interventions.
- Role clarity helps navigate the system effectively, saving educators' time.

3.2 Course Management Module

This module facilitates the creation, enrollment, and management of courses.

3.2.1 Course Creation and Management (Teacher)

Description: Teachers can create, update, and delete courses, assign schedules, and manage course content.

How it Improves Teaching Methods:

- Facilitates the structuring of courses by teachers appropriately, allowing for relevant content for students.
- Facilitates better planning and organization, improving overall learning outcomes.

3.2.2 3.2.2 Enrollment (Student)

Description: Students can browse and enroll in courses. Teachers approve enrollments, ensuring the right participants.

How it Enhances Teaching Strategies:

- Ensures students are enrolled in relevant courses, maintaining a focused learning environment.
- Provides teachers with control over participant selection, enabling personalized engagement.

3.3 Live Class Facilitation Module

This module supports real-time interaction and monitoring.

3.3.1 Virtual Classes

Description: Video streaming, screen sharing, and real-time interaction tools are offered to the students.

How it Enhances Teaching Strategies:

- Interactive sessions enhance engagement and participation.
- Immediate feedback is possible from the teacher, who can gauge the reaction from the students.

3.3.2 Expression Detection

Description: AI monitors students' expressions in real-time using Python machine learning models.

How it Enhances Teaching Strategies:

- Gives teachers instantaneous feedback on students' emotions about being confused or disinterested.
- Encourages adaptive teaching techniques based on student needs.

3.4 Reporting Module

This module provides attendance and engagement reports.

3.4.1 Attendance Reports

Description: Tracks the number of students attending live sessions in tabular form.

How it Enhances Teaching Strategies:

- Points out students with poor attendance so that follow-up may be made accordingly.
- Ensures accountability and helps teachers maintain class discipline.

3.4.2 Emotion Detection Reports

Description: Supplies post-class engagement analysis with graphical summaries of student emotions.

How it Improves Teaching Approaches:

- Helps teachers know how effective their approach is.
- Shows student engagement trends, which might be helpful in future teaching approaches.

4 Model Training and Evaluation

4.1 Data Augmentation and Training Process

The model was trained using data augmentation techniques to prevent overfitting and enhance generalization. These techniques included:

- **Rescaling:** Images were rescaled to a [0,1] range to standardize the pixel values.
- **Random Rotations:** Random rotations of the images by a small angle were applied to make the model invariant to rotations.
- **Translation and Shearing:** Random translations and shearing of the images were applied to help the model generalize better.
- **Horizontal Flipping:** Horizontal flipping was used to help the model learn to detect facial expressions from different angles.

4.2 Model Architecture

The model used a Convolutional Neural Network (CNN) for image classification. The CNN architecture used in the project consists of the following layers:

- **Convolutional Layers:** These layers detect important features from the input images.
- **Max-Pooling Layers:** These layers reduce the spatial dimensions of the feature maps, which helps in lowering computation and prevents overfitting.
- **Fully Connected Layers:** These layers are responsible for classifying the features extracted by the convolutional layers into one of the predefined classes.
- **Dropout Layers:** These layers are used to reduce overfitting by randomly dropping units during training.

4.3 Performance Evaluation

The performance of the model was evaluated on the training dataset using classification metrics such as accuracy, precision, recall, and F1-score. These metrics are used to evaluate the effectiveness of the model, especially in the context of an imbalanced dataset.

4.4 Classification Report and Model Evaluation Metrics

The following table shows the classification report for the model's performance:

Table 1: Model Classification Report

Emotion	Precision	Recall	F1-Score	Support
Angry	0.14	0.12	0.13	3995
Disgust	0.00	0.00	0.00	436
Fear	0.14	0.06	0.08	4097
Happy	0.25	0.28	0.26	7215
Neutral	0.18	0.24	0.20	4965
Sad	0.17	0.18	0.18	4830
Surprise	0.11	0.12	0.11	3134
Accuracy		0.18		28672
Macro Avg	0.14	0.14	0.14	28672
Weighted Avg	0.17	0.18	0.17	28672

4.5 Explanation of Evaluation Metrics

- **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is the ratio of the correct predictions to the total predictions. However, it may not always be the best measure in imbalanced datasets.

- **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision indicates the proportion of true positive predictions among all positive predictions made by the model.

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall measures how many of the actual positive instances were correctly predicted.

- **F1-Score:**

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score is the harmonic mean of precision and recall. It is a balanced metric that takes both false positives and false negatives into account.

5 Wireframes

5.1 Landing Page

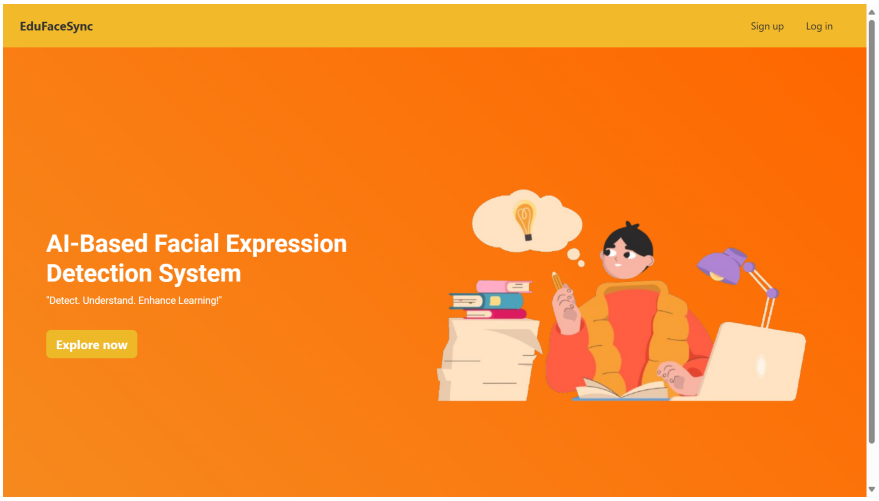

Figure 1: Main Page

5.2 Sign-Up Page

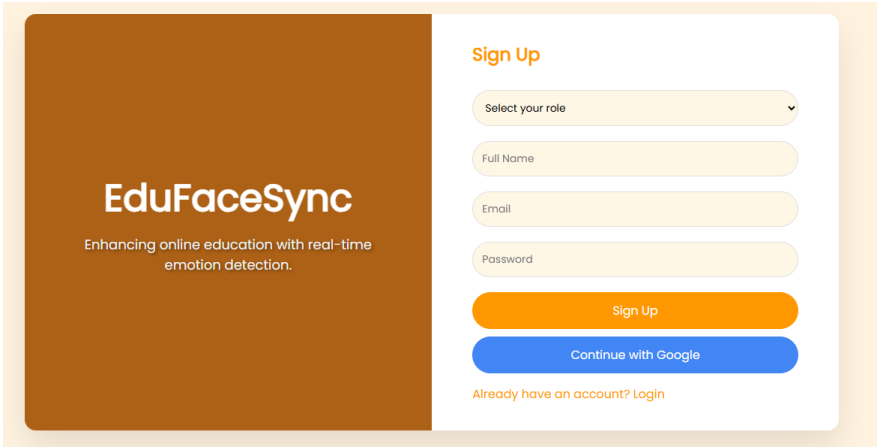

Figure 2: Sign-up page

5.3 Login Page

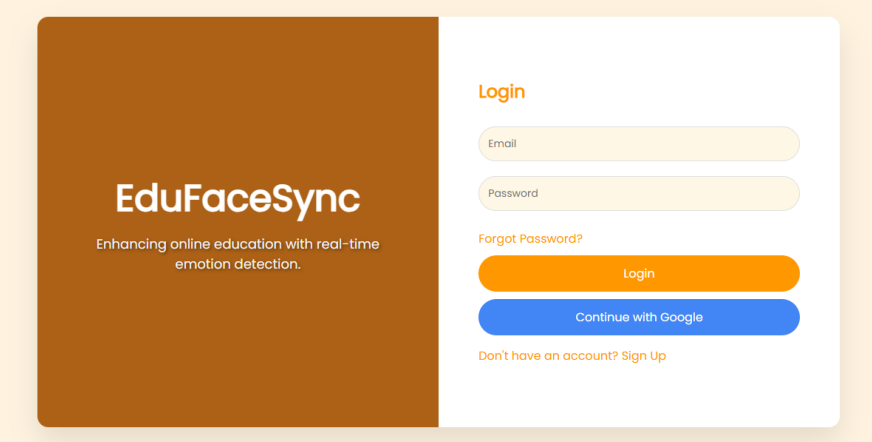

Figure 3: Login-page

5.4 Reset Password

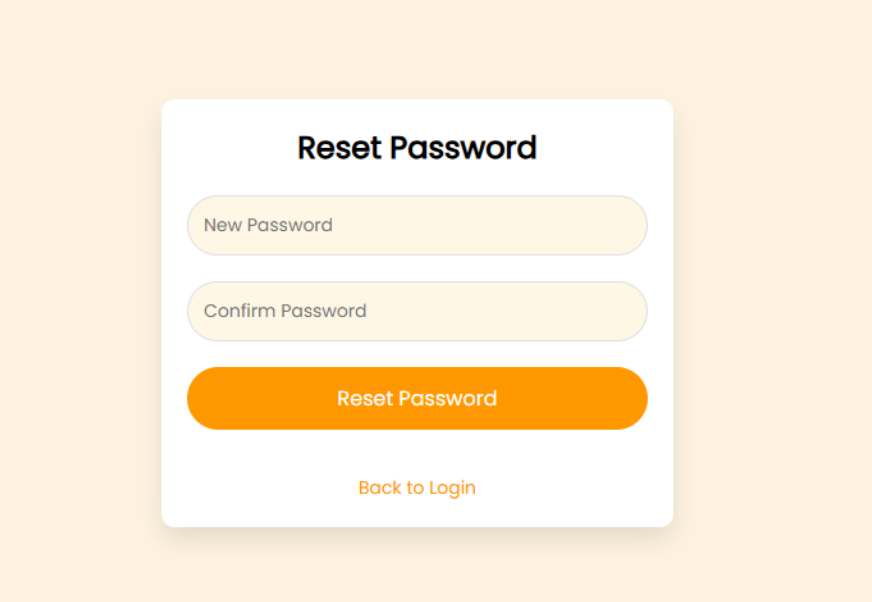

Figure 4: Reset password

5.5 Teacher Dashboard

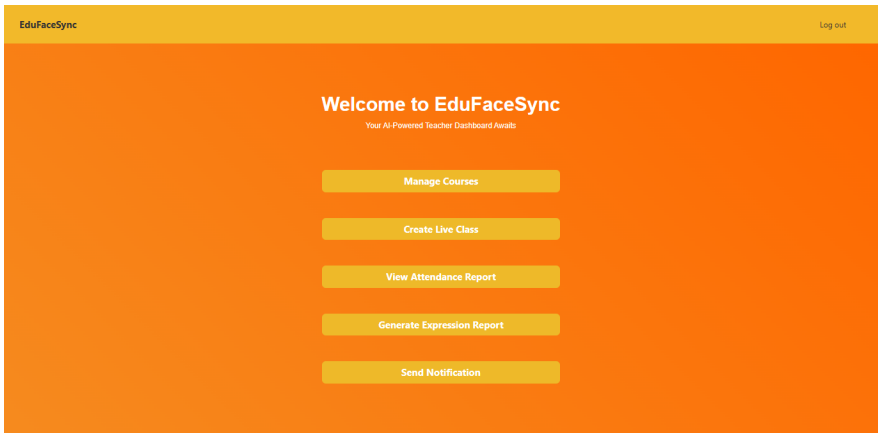

Figure 5: Teacher dashboard

5.6 Manage Course Page

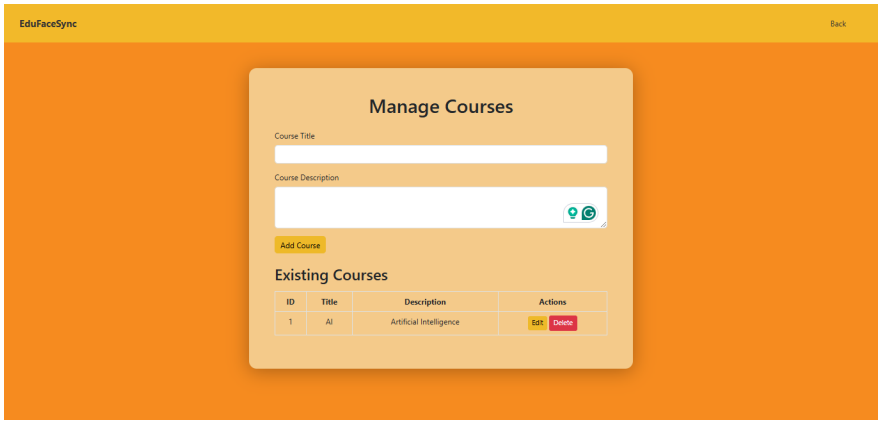

Figure 6: Add,Update, Delete existing course

5.7 Create Live Class Page

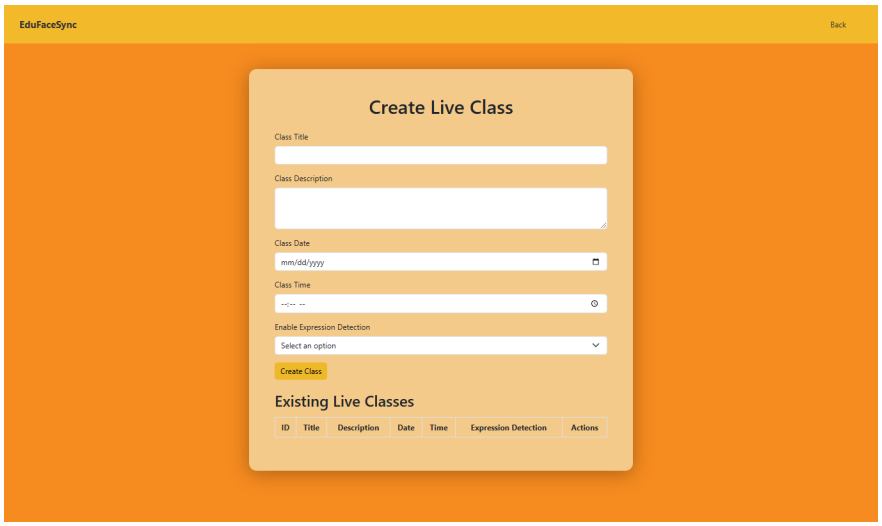

Figure 7: Create live class

5.8 Attendence Report Page

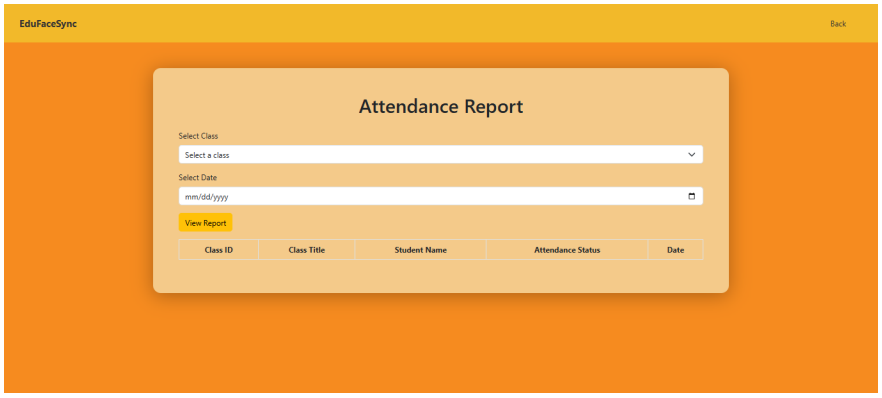

Figure 8: View Attendance report

5.9 Expression Report Page

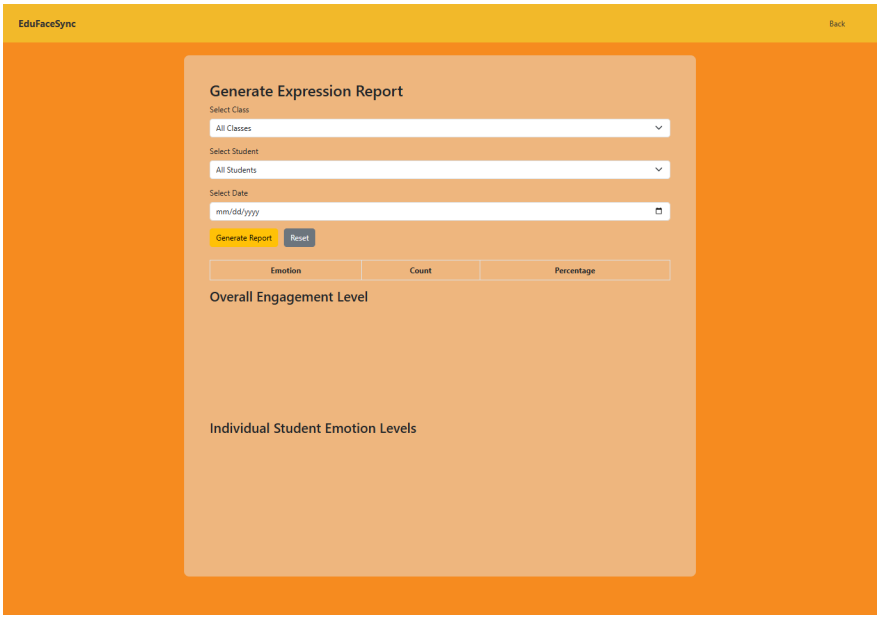

Figure 9: Expression Report

5.10 Student Dashboard Page

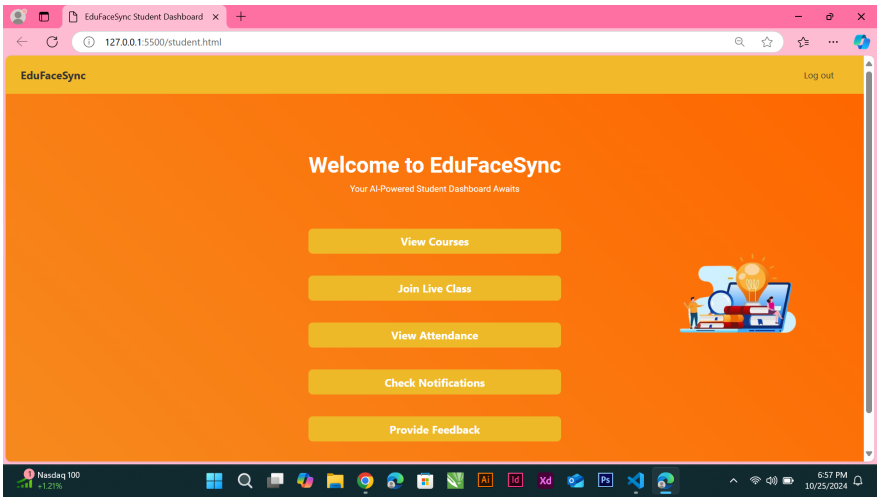

Figure 10: Student Dashboard

5.11 View Available Course Page

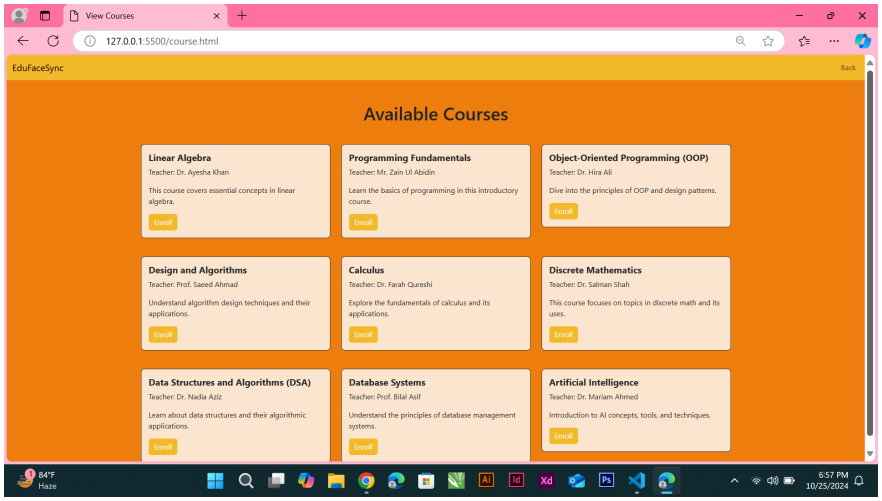

Figure 11: Student can view All available Courses

5.12 Join Live class Page

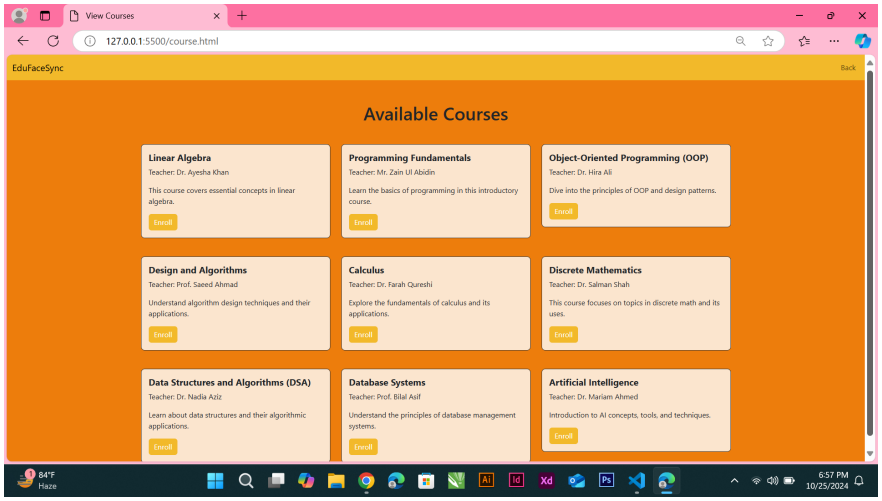

Figure 12: Join Live class

5.13 View Attendance Check Page

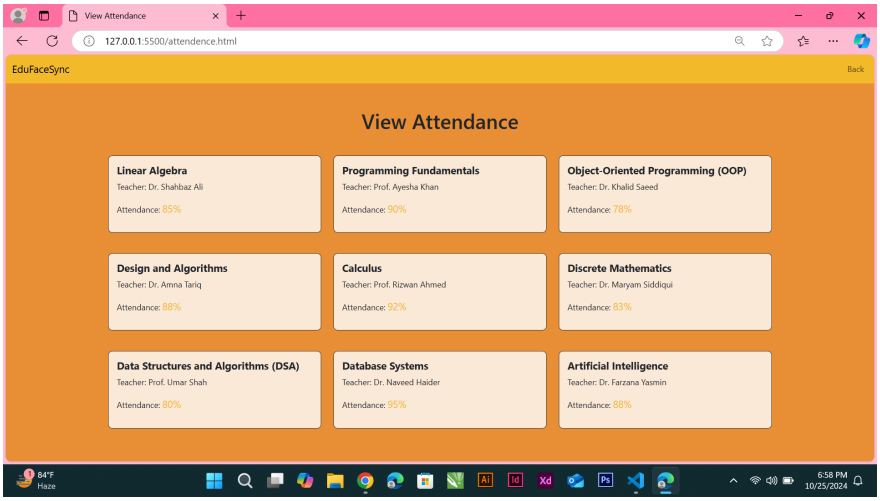

Figure 13: View Attendance

6 Functional Requirements of EduFaceSync System

The following sections describe the working flow of the EduFaceSync system based on the functional requirements.

6.1 User Sign-Up

1. **User Accesses Sign-Up Page:** The user navigates to the sign-up page of EduFaceSync.
2. **Enter User Details:** The user provides their email, password, and selects their role (either teacher or student).
3. **Form Validation:** The system validates the input fields (e.g., correct email format, password strength, and role selection).
4. **Account Creation:** Upon successful validation, the system creates a new account.
5. **Confirmation Email:** The system sends a confirmation email to the user.
6. **User Receives Confirmation:** The user receives a notification confirming successful sign-up.
7. **Error Handling:** If there are any errors (e.g., invalid email, role not selected, or duplicate email), the system prompts the user with an error message and asks for correction.

6.2 User Login

1. **User Accesses Login Page:** The user navigates to the login page.

2. **Enter Credentials:** The user enters their registered email and password.
3. **Form Submission:** The system processes the login request by validating the email and password.
4. **Role Selection:** The user's role (teacher or student) is validated, and they are directed to the appropriate dashboard.
5. **Login Success:** If the credentials are valid, the user is logged in and redirected to their respective dashboard.
6. **Error Handling:** If invalid credentials are provided, the system prompts the user to re-enter their information. If the user forgot their password, the system provides a password reset option.

6.3 Course Creation (Teacher)

1. **Teacher Accesses Course Creation Page:** The teacher navigates to the course creation page.
2. **Enter Course Details:** The teacher provides relevant course details such as title, description, schedule, and course material.
3. **Form Validation:** The system validates the input, ensuring no missing required fields (e.g., course title, description).
4. **Course Creation:** If all fields are correct, the system creates the course and stores it in the system.
5. **Confirmation:** The system sends a confirmation notification to the teacher indicating that the course has been successfully created.
6. **Error Handling:** If the course already exists or if required fields are missing, the system provides an error message and prompts the teacher to correct the issue.

6.4 Course Enrollment (Student)

1. **Student Accesses Course Enrollment Page:** The student navigates to the course enrollment page.
2. **View Available Courses:** The system displays a list of available courses for enrollment.
3. **Select and Enroll:** The student selects a course they wish to enroll in and clicks on the "Enroll" button.
4. **Enrollment Processing:** The system processes the enrollment request, checking the availability of the course.
5. **Confirmation:** The system confirms the student's enrollment and sends a notification to the student and the respective teacher.

6. **Error Handling:** If the course is full or if the enrollment deadline has passed, the system notifies the student and offers alternatives such as waitlisting or future courses.

6.5 Live Class Creation (Teacher)

1. **Teacher Accesses Live Class Creation Page:** The teacher navigates to the live class creation page.
2. **Enter Class Details:** The teacher specifies the date, time, and duration for the live class.
3. **Form Validation:** The system checks for conflicts in date and time and validates that the date/time is in the future.
4. **Generate Link:** Upon successful validation, the system generates a unique link for the live class.
5. **Notification:** The system sends the generated link to enrolled students.
6. **Error Handling:** If the date or time is invalid (e.g., in the past), or if there is a scheduling conflict, the system notifies the teacher with an error message.

6.6 Student Joins Live Class

1. **Student Receives Notification:** The student receives a notification about the upcoming live class with a link to join.
2. **Join Class:** The student clicks the provided link to join the live class.
3. **System Verification:** The system verifies that the student is enrolled in the class.
4. **Join Success:** If the student is enrolled, they are granted access to the live class.
5. **Update Attendance:** The system updates the attendance record to mark the student as present.
6. **Error Handling:** If the link has expired or if there are technical issues, the system displays an error and offers the student alternatives (e.g., contact the teacher).

6.7 Automatic Attendance Tracking

1. **Monitor Participation:** The system automatically monitors student participation during the live class by detecting when they join the session.
2. **Record Attendance:** The system records the attendance of each student present during the session.
3. **Update Database:** The system updates the attendance database with the list of students who attended the class.

4. **Error Handling:** In case of disconnections or system errors, the system flags the attendance record for review.

6.8 Emotion Detection Report Generation

1. **Capture Facial Expressions:** During the live class, the system detects and analyzes the facial expressions of students using emotion detection algorithms.
2. **Generate Report:** After the class, the system generates an emotion detection report based on the students' facial expressions during the class.
3. **Send Report to Teacher:** The system sends the generated report to the respective teacher.
4. **Error Handling:** If the system fails to detect expressions or if there is a report generation error, the teacher is notified, and an alternative (manual report) is suggested.

7 Non-Functional Requirements

The following non-functional requirements have been defined for the EduFaceSync system:

- **Performance:** The system shall respond to user inputs with minimal latency, ensuring that operations like sign-up, login, course enrollment, and class attendance tracking are completed within 3 seconds.
- **Scalability:** The system shall support up to 50 concurrent users without degradation in performance. As the user base grows, the system should be able to handle increased load by scaling resources dynamically.
- **Availability:** The system shall be available 24/7, with a downtime of no more than 4 hours annually for maintenance and updates.
- **Reliability:** The system should have an uptime of 70%, ensuring continuous availability of features such as real-time emotion detection and live class functionalities.
- **Security:** The system shall implement security measures to protect user data, including encrypted passwords, secure login processes, and HTTPS connections for all interactions.
- **Usability:** The system interface should be user-friendly, with a simple design for both students and teachers, ensuring that they can navigate the platform easily without the need for extensive training.
- **Maintainability:** The system shall be easy to maintain and update, with well-documented code and an efficient structure that allows for quick bug fixes, updates, and new feature additions.

- **Compatibility:** The system shall be compatible with commonly used browsers (e.g., Chrome, Firefox, Edge) and mobile devices, ensuring that both students and teachers can access the platform without issues.
- **Compliance:** The system shall comply with applicable data privacy regulations (e.g., GDPR) to protect user information, especially concerning the collection of facial expression data and personal details.
- **Support for Future Growth:** The system should be designed with the capability to integrate additional features in the future, such as support for multi-language and expanded emotion recognition functionalities.

8 Emotion Detection Model Using Convolutional Neural Networks (CNN)

The emotion detection model in this project is a Convolutional Neural Network (CNN) designed to classify facial expressions into seven categories: Angry, Disgusted, Fearful, Happy, Neutral, Sad, and Surprised. The model leverages a deep learning approach to analyze facial features and predict the emotional state of an individual.

8.1 Model Architecture

The model is built using the Sequential API from Keras, which is part of the TensorFlow library. The architecture consists of several convolutional layers, max-pooling layers, and dense layers. Below is a detailed explanation of each layer:

1. **Convolutional Layers:** The first three layers are convolutional layers (Conv2D) with ReLU (Rectified Linear Unit) activation. These layers automatically extract features from the input images (which are 48x48 pixels in size). The number of filters increases as we move deeper into the network:
   - The first layer has 64 filters, with a kernel size of 3x3.
   - The second layer has 128 filters, with a kernel size of 3x3.
   - The third layer also has 128 filters, with a kernel size of 3x3.

   These convolutional layers help detect basic features such as edges and textures in the images.
2. **Max-Pooling Layers:** Max-pooling (MaxPooling2D) layers are used after each convolutional layer to reduce the spatial dimensions of the feature maps. This helps in reducing the computational load and extracting higher-level features. The pool size is set to 2x2, which means the spatial dimensions are halved after each pooling layer.
3. **Flatten Layer:** The flatten layer is used to reshape the output from the convolutional layers into a 1D vector, making it suitable for the dense layers that follow.

4. **Dense Layers:** The first dense layer has 512 units and uses ReLU activation. This layer acts as a fully connected layer that processes the features extracted by the convolutional layers. The second dense layer is the output layer, which consists of 7 units (one for each emotion class), and uses the softmax activation function. The softmax function converts the raw output values into probabilities, with the highest probability indicating the predicted emotion.
5. **Dropout:** A dropout rate of 0.5 is applied to the first dense layer. Dropout is a regularization technique used to prevent overfitting during training. It works by randomly setting some of the neurons to zero during each training iteration, forcing the network to learn more robust features.

The model is compiled using the following parameters:

- **Loss Function:** Categorical Crossentropy is used as the loss function, which is suitable for multi-class classification problems.
- **Optimizer:** The Adam optimizer is used with a learning rate of 0.0001. Adam is a popular optimizer due to its adaptive learning rate capabilities.
- **Metrics:** The accuracy of the model is monitored during training.

8.2 Data Augmentation

To improve the model's generalization ability and prevent overfitting, data augmentation is applied to the training images. The following transformations are applied to each image:

- **Rotation:** Images are randomly rotated within a range of 40 degrees.
- **Width and Height Shifts:** Images are randomly shifted horizontally and vertically by 20
- **Shear:** A shear transformation is applied with a 20
- **Zoom:** Images are randomly zoomed in or out by 20
- **Horizontal Flip:** Images are randomly flipped horizontally.

This helps create more varied training data, improving the model's robustness.

8.3 Model Training and Evaluation

The model is trained using the 'fit' method on a dataset of facial images that have been labeled with one of the seven emotions. During training, the model's weights are updated to minimize the categorical crossentropy loss using the Adam optimizer. The training process includes monitoring the model's performance on the training set, and the best model is saved using the 'ModelCheckpoint' callback.

After training, the model is evaluated using a confusion matrix and classification report. The following evaluation metrics are calculated:

- **Accuracy:** The percentage of correct predictions made by the model.
- **Precision, Recall, F1-Score:** These metrics are calculated for each class to evaluate the model's performance in terms of both false positives and false negatives.

8.4 Model Usage for Real-Time Emotion Detection

The trained model is used for real-time emotion detection through a webcam feed. The steps involved are:

1. **Face Detection:** A Haar Cascade Classifier is used to detect faces in the video feed.
2. **Preprocessing:** For each detected face, the region of interest (ROI) is extracted, resized to 48x48 pixels, and converted to grayscale.
3. **Emotion Prediction:** The preprocessed face image is passed to the trained CNN model, which predicts the emotion label.
4. **Display Prediction:** The predicted emotion label is displayed on the video feed next to the detected face.

This allows the model to predict emotions in real-time based on facial expressions detected by the webcam.

9 Conclusion

EduFaceSync combines state-of-the-art technologies like Python, MySQL, and WebRTC to revolutionize online education. It empowers teachers to refine their teaching strategies, enhance student engagement, and foster a more effective learning environment by providing real-time insights and detailed analytics. Future updates could include multi-language support, advanced analytics, and integration with third-party LMS platforms.