JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AI – ENHANCED REMOTE EXAM MONITORING WITH COMPUTER VISION TECHNIQUES

[1]Santhi, [2]Abhi Rama Sarma, [3]N. Saikrishna, [4]P. Naresh, [5]P. Dhinakar
[1]Assistant Professor , [2,3,4,5]B.Tech Student
[1,2,3,4,5]Department of Computer Science and Engineering ,Narayana Engineering College, Nellore, Andhra Pradesh, India

Abstract : The COVID-19 pandemic has accelerated the shift to online learning, highlighting the need for innovative strategies to maintain academic integrity in virtual exams. This project introduces an advanced Proctored Online Examination System using deep learning and computer vision technologies to enhance security and efficiency. The system supports faculty and students, managed by super admins. Faculty members log in manually, while students register and authenticate securely. The student dashboard provides access to courses and exams, featuring a compiler for coding practice in C, C++, SQL, Python, and Java. Sophisticated deep learning algorithms, including OpenCV's LBPH (Linear Binary Pattern Histogram), are used for face detection and recognition, ensuring continuous identity verification during exams. Initial testing shows high accuracy in verifying student identities, reducing cheating. This system offers a secure, interactive environment for online exams, supporting robust identity verification and dynamic learning. It has the potential to revolutionize online education by ensuring exam integrity, fostering greater trust in online learning platforms, and promoting equitable access to education.

***IndexTerms* - Online Learning, Proctored Exams, Deep Learning, Computer Vision, Academic Integrity, Face Recognition, LBPH, Identity Verification, OpenCV, Interactive Learning**

I. INTRODUCTION

In recent years, education has seen a substantial transformation, particularly accelerated by the COVID-19 pandemic. This shift has made the swift adoption of online learning platforms essential, bringing both opportunities and challenges in maintaining academic integrity during virtual assessments. Traditional examination methods are no longer viable, prompting institutions to seek innovative solutions for secure and fair testing environments.

Our project, the Online Examination System using Deep Learning and Computer Vision, addresses these challenges by integrating advanced technologies to create a robust and reliable proctoring system. At its core, our system uses deep learning algorithms and computer vision techniques, particularly facial recognition, to authenticate and monitor students during exams. The system comprises two primary user roles: faculty and students, managed by super admins. Super admins oversee the entire operation, managing faculty login data and ensuring the smooth functioning of the platform. Faculty members can log into their accounts and access an admin dashboard to manage courses and exams. They can add new courses by submitting relevant details for super admin approval. Additionally, faculty can schedule exams, upload questions via Excel sheets, and monitor student performance by accessing detailed reports.

For students, the process begins with registration and setting up login credentials. Upon successful login, students are directed to a personalized dashboard where they can access courses for practice, integrated with a compiler developed using Node.js Compilex. The examination process employs OpenCV in Python and the Linear Binary Pattern Histogram (LBPH) algorithm to perform facial recognition. This ensures that the student attending the exam is indeed the registered individual. If the facial match is successful, the exam questions are displayed, and the student's attendance is marked. Otherwise, an error message is generated, preventing unauthorized access.

Our system not only upholds the integrity of online exams but also enhances the user experience by providing a seamless and secure environment for both students and faculty. By leveraging cutting-edge deep learning and computer vision technologies, our project sets a new standard for online examination systems, ensuring that educational institutions can continue to deliver assessments with confidence and accuracy in a virtual setting.

II. MOTIVATION OF THE PROJECT

The rapid shift to online learning prompted by the COVID-19 pandemic has fundamentally changed the educational landscape. Traditional in-person examinations have become impractical, driving the need for innovative solutions that ensure the integrity and fairness of virtual assessments. This project, an Online Examination System utilizing Deep Learning and Computer Vision, emerges as a response to this urgent need. Ensuring academic integrity in a virtual environment is a significant challenge. Conventional online examination systems often fall short in preventing dishonest practices, such as impersonation or unauthorized assistance. Our system addresses these issues head-on by incorporating advanced technologies for secure and reliable proctoring. By leveraging deep learning algorithms and computer vision techniques, we provide robust face detection and recognition capabilities, which are critical in authenticating the identity of students during exams.

The core motivation behind this project is to create a seamless, secure, and efficient online examination process. For students, the system offers a user-friendly interface that integrates an online compiler, enabling them to practice coding and take exams within the same platform.

This holistic approach not only streamlines their learning experience but also ensures that they can demonstrate their knowledge in a controlled and monitored environment. For faculty, the system provides a comprehensive dashboard for managing courses and exams. Faculty members can add new courses, schedule exams, and upload questions easily, minimizing the administrative burden. Additionally, the ability to access detailed student performance reports enables faculty to track progress and identify areas where students may need additional support.

Super admins play a crucial role in maintaining the system's integrity by overseeing faculty login data and approving new courses. This hierarchical structure ensures that there is always a layer of oversight, enhancing the system's overall security and reliability. The use of OpenCV in Python and the Linear Binary Pattern Histogram (LBPH) algorithm for facial recognition adds a sophisticated layer of security, ensuring that only registered students can access exam questions. This approach not only prevents impersonation but also builds trust in the online examination process.

Abbreviations and Acronyms

- **COVID-19** - Coronavirus Disease 2019
- **CV** - Computer Vision
- **LBPH** - Linear Binary Pattern Histogram
- **SQL** - Structured Query Language
- **OpenCV** - Open Source Computer Vision Library
- **Node.js** - Node JavaScript
- **SSL** – Secure Socket Layer

Equations

Local Binary Pattern (LBP) is a straightforward yet highly effective texture operator used to label pixels in an image by comparing each pixel with its neighbors. Each pixel is thresholded against its surrounding neighbors, and the result is considered a binary number.

By using LBP combined with histograms, we can effectively represent face images as simple data vectors, making it a useful tool for face recognition tasks. Below is a step-by-step explanation of the LBPH algorithm for face recognition.

Parameters of the LBPH Algorithm

The LBPH algorithm uses four primary parameters:

1. **Radius**: The radius defines the circular area around each pixel. Typically, the radius is set to 1.
2. **Neighbors**: This parameter specifies the number of sample points around each pixel to form the circular LBP. Commonly, it is set to 8. Increasing the number of neighbors increases computational complexity.
3. **Grid X**: This parameter determines the number of cells in the horizontal direction of the image grid. More cells result in a finer grid and higher feature vector dimensionality. The usual setting is 8.
4. **Grid Y**: This parameter determines the number of cells in the vertical direction of the image grid. As with Grid X, more cells result in a finer grid and higher feature vector dimensionality. The usual setting is 8.

These parameters are crucial for the proper functioning of the LBPH algorithm, and their optimal values depend on the specific application and dataset.

Training the LBPH Algorithm

To train the LBPH algorithm, a dataset of facial images with corresponding IDs (either numbers or names) is needed. Each ID should be unique to the individual in the images. The training process involves computing the LBP for each image and storing the resulting histograms.

Applying the LBP Operation

The first step in the LBPH algorithm is to create an intermediate image that better highlights facial features. This is done using a sliding window approach, based on the radius and neighbors parameters.

Steps:

Grayscale Image: Convert the facial image to grayscale.

1. **3x3 Window**: For each pixel, consider a 3x3 window of neighboring pixels.
2. **Thresholding**: Use the central pixel value as a threshold. Compare each neighboring pixel value with the threshold.
3. **Binary Pattern**: Assign binary values (1 or 0) based on whether neighboring pixel values are above or below the threshold.
4. **Binary to Decimal**: Convert the binary pattern into a decimal value and assign it to the central pixel.
5. **Resultant Image**: After processing all pixels, the resultant image will have enhanced facial features.

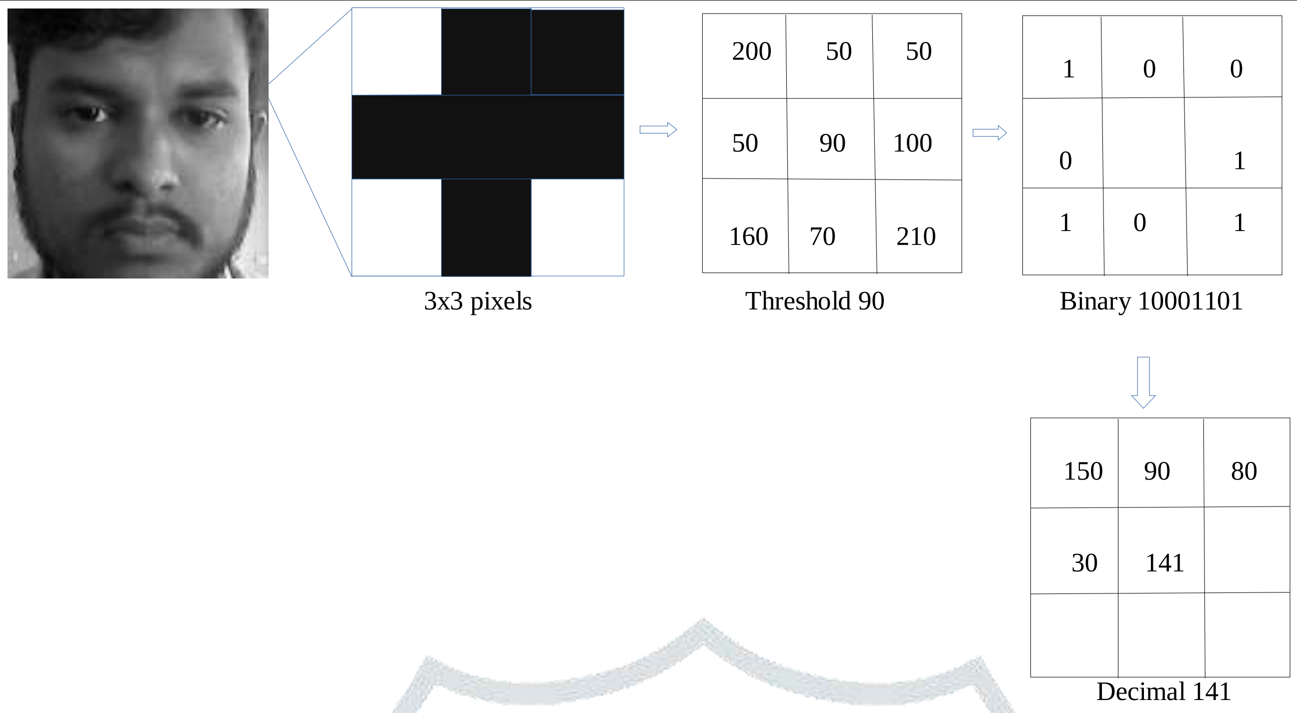


fig a : LBP operation

This process can be extended to circular LBP by using bilinear interpolation to handle non-integer pixel positions.

Extracting Histograms

After applying LBP, the next step is to divide the image into multiple grids (defined by Grid X and Grid Y parameters). For each grid cell, a histogram of pixel intensities (0 to 255) is computed.

Steps:

1. **Grids**: Divide the image into grids.
2. **Histograms**: Compute the histogram for each grid cell.
3. **Concatenation**: Concatenate all histograms to form a single feature vector representing the entire image.

Face Recognition Using LBPH

To recognize a face, the algorithm compares the feature vector of the input image with those in the training dataset. The comparison is done by measuring the distance between histograms using methods like Euclidean distance.

Steps:

1. **Feature Extraction**: Apply LBP and extract the histogram for the input image.
2. **Distance Calculation**: Compute the distance between the input image's histogram and each histogram in the training
3. dataset.

Euclidean Distance for Histogram Comparison

Given two histograms H1H_1H1 and H2H_2H2 of length n:

$$d(H_1,H_2)=\sqrt{\Sigma^n{}_{i=1}}(H_1[i] - H_2[i[]])^2 \qquad (1)$$

Where d(H1,H2)d(H_1, H_2)d(H1,H2) is the Euclidean distance between histograms H1H_1H1 and H2H_2H2.

4. **Best Match**: The ID of the closest match (smallest distance) is returned as the recognized face. The algorithm also provides a confidence score, with lower scores indicating better matches.

The confidence score can be used to set a threshold for recognition accuracy, ensuring that only highly confident matches are accepted.

II. PROPOSED WORK

System Overview

The proposed Online Examination System harnesses the power of deep learning and computer vision to facilitate secure and efficient online exams. This system features two primary users: faculty and students. Super admins oversee the system, managing faculty login data, which is stored in the admin_users table of the database, while students register themselves, and their login details are stored in the student_users table. Both faculty and students have login credentials that include an email and password.

Student Roles and Dashboard

When students log in, their credentials are verified, and they are redirected to the student dashboard, which offers two main tasks: courses and exams.

Courses:

Students can access and practice courses using an integrated compiler.

The compiler supports multiple programming languages, including C, C++, SQL, Python, and Java.

Exams:

Students can select and attend scheduled exams listed in their dashboard.

Exam details are retrieved from the exams table in the database.

Before starting an exam, students must undergo facial attendance marking using OpenCV in Python and the LBPH (Linear Binary Pattern Histogram) algorithm.

If the student's face matches the trained data stored in the trainer.yml file, attendance is marked, and the exam is opened.

Exam questions are fetched from the questions and options tables in the database.

For coding exams, students use a compiler developed with Node.js Compilex, supporting Python and Java.

Upon completion, exam results are stored in the results table in the database.

Faculty Roles and Dashboard

Faculty members, upon successful login, access the admin dashboard, where they have three primary tasks:

Add Course:

Faculty can add a course by submitting a form with required information such as course name, course level (easy/medium/hard), expected time to complete, and contact information.

This form data is sent to the super admin, who then approves and adds the course to the system.

Add Exam:

Faculty can add exams by submitting form data including course, subject, question level (easy/medium/hard), number of questions, exam duration, and start time.

Based on this information, questions are fetched from the questions table and scheduled in the exams section for students.

Faculty can also upload exam questions via an Excel sheet, which is then processed and stored in the database.

Reports:

Faculty can view detailed reports of student performance by searching for a student using their ID number.

The results are fetched from the results table and displayed for the faculty to review.

Key Technologies and Their Relevance

Node.js Compilex:

Definition: Compilex is a Node.js module that enables the compilation and execution of code within a web application.

Uses: It supports the execution of coding exams in multiple languages such as Python and Java.

Relevance: By integrating Compilex, the system provides a real-time coding environment, allowing students to write and test code within the platform, ensuring a seamless and practical exam experience.

OpenCV:

Definition: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library.

Uses: Used for facial recognition to verify the identity of students before they access their exams.

Relevance: OpenCV is crucial for implementing robust and accurate facial recognition, ensuring that only authorized students can attend the exams, thus maintaining the integrity of the examination process.

LBPH (Linear Binary Pattern Histogram):

Definition: LBPH is a simple yet efficient face recognition algorithm that works with local binary patterns.

Uses: Utilized in facial recognition for comparing the student’s face with the trained data stored in the trainer.yml file.

Relevance: LBPH provides reliable facial recognition even in varying lighting conditions, ensuring consistent and accurate attendance marking for students.

System Security and Integrity

The proposed system incorporates multiple layers of security and integrity measures:

Authentication: Secure login mechanisms for students and faculty to prevent unauthorized access.

Facial Recognition: Ensures that the student taking the exam is the registered individual, preventing impersonation.

Data Management: Secure handling of all data, including exam details, student performance, and course information.

Real-time Monitoring: Continuous monitoring during exams to detect and prevent suspicious behavior.

By leveraging advanced technologies such as deep learning, computer vision, OpenCV, and LBPH, the proposed system aims to provide a secure, efficient, and reliable online examination platform, addressing the challenges posed by traditional examination methods in a virtual learning environment.

III. WORKING

Our project is an extensive online examination system that harnesses the power of deeplearning and computer vision technologies.It comprises two pivotal user roles: faculty and student.Super admins wield authority over all faculty login data, whereas students autonomouslyregister for access. Login credentials, encompassing email and password, are securelystored in the admin_users and student_users tables within the database, respectively.

Student Roles:

Upon logging in, students are seamlessly directed to their personalized dashboard,offering two primary tasks: Courses and Exams.

In the Exams section, students are tasked with the responsibility of verifying their attendance through facial recognition, a sophisticated feature facilitated by OpenCV's LBPH (Linear Binary Pattern Histogram) algorithm. Only upon successful attendance marking are students granted access to their scheduled exams. Further enhancing the educational experience, the Courses section is equipped with an ntegrated compiler supporting an array of programming languages, including C, C++, SQL, Python, and Java. This compiler empowers students to practice and refine their coding skills in a dynamic and interactive environment. Scheduled exams are meticulously organized and displayed, seamlessly sourced from the database's comprehensive exams table.

Faculty Roles:

Upon successful login, faculty members are ushered into the administrative domain,
where they wield an array of pivotal tasks:

Add Course:

Faculty members possess the capability to augment the course catalog by submitting ameticulously curated form containing vital course details such as name, difficulty level(ranging from easy to medium to hard), anticipated time to receive the code, and contactphone number.This form data is efficiently relayed to super admins via email, streamlining the processof course addition.

Add Exam:

Faculty members exercise their authority by meticulously crafting exams through astreamlined submission process.This entails furnishing essential form data, including course, subject, question difficultylevel (ranging from easy to medium to hard), number of questions, time duration, and start time. Leveraging this data, questions are meticulously curated from the database's expansive questions table and seamlessly disseminated as a comprehensive question paper to students' scheduled exams sections.

View Reports:

Faculty members are bestowed with the invaluable ability to peruse student reports withconsummate ease. Through a seamless search mechanism utilizing the student's unique identification number, pertinent data is meticulously retrieved from the results table within the database and elegantly presented to the faculty member.

Key Technologies:

OpenCV: This indispensable technology serves as the cornerstone for facial recognitionbased attendance marking, underpinned by the sophisticated LBPH (Linear Binary Pattern Histogram) algorithm. OpenCV ensures robust identity verification, fortifying the integrity of the examination process.

LBPH (Linear Binary Pattern Histogram): LBPH emerges as a pivotal component for facial recognition and authentication, facilitating the seamless and secure attendance marking process.

Compilex: A cutting-edge compiler ingeniously developed utilizing Node.js, serving as a versatile platform supporting both Python and Java programming languages. This compiler engenders an enriching educational experience, fostering the development and refinement of students' programming prowess.

Project Relevance:

Facial Recognition-based Attendance: OpenCV is crucial for implementing facial recognition-based attendance marking. By utilizing OpenCV's capabilities, the system can verify the identity of students logging in for exams. This enhances security and ensures that only authorized individuals can access exam materials.

Enhanced Security: OpenCV's facial recognition technology adds an additional layer of security to the examination system, safeguarding against unauthorized access and impersonation.

Efficient Authentication: OpenCV streamlines the authentication process by automating attendance marking, saving time for both students and faculty. This automation contributes to the overall efficiency of the examination system.

Compilex: Support for Multiple Programming Languages, Compilex enables students to practice coding skills in languages such as Python and Java, which are relevant to various courses and exams within the system. This broad language support accommodates the diverse needs of students pursuing different programming disciplines.

Dynamic Learning Environment: By providing an integrated compiler, the system offers a dynamic learning environment where students can write, compile, and test code in real-time. This hands-on approach fosters active learning and skill development.

Assessment Capabilities: Compilex facilitates coding exams within the system, allowing faculty to assess students' programming proficiency effectively.Students can complete coding tasks and receive immediate feedback, enhancing the assessment process's efficiency and transparency.

Server Hosting Process: Creating Flask Application, we started by developing a Flask application for attendance marking and proctoring. Flask is a Python web framework that allows you to build web applications. You write the necessary code to handle attendance marking and proctoring functionalities.

Installing mkcert: Mkcert is a tool used for generating SSL certificates for localhost development. You install mkcert on our system if you haven't already.

Generating SSL Certificate: Using mkcert, you generate an SSL certificate for our localhost. This certificate is required to enable HTTPS on your local server.

Configuring Flask for HTTPS: You configured our Flask application to serve over HTTPS by providing the SSL certificate and key generated by mkcert.

Testing Locally: we tested our Flask application locally over HTTPS to ensure everything is working as expected.

Using Ngrok: Ngrok is a tool that creates a secure tunnel to our localhost, allowing you to expose your local server to the internet. You start Ngrok and point it to our Flask application running on localhost.

Connecting Ngrok to Flask: Ngrok generates a temporary public URL (usually in the form of https://randomstring.ngrok.io) that tunnels to our localhost. You connect this Ngrok URL to our Flask application.

Accessing Application: Now, anyone with access to the Ngrok URL can access your Flask application over the internet, with the added security of HTTPS. This setup allows us to develop and test your Flask application locally with HTTPS, and then make it accessible to others over the internet using Ngrok. It's a common workflow for developing and showcasing web applications before deploying them to a production server.

IV. RESULTS AND DISCUSSION

4.1 Results of Descriptive Statics of Study Variables

Innovative Solutions for Overfitting in Smart Proctoring Systems: Discussions revolve around introducing advanced deep learning models, such as Convolutional Neural Networks (CNN), to address overfitting challenges in online proctoring. Leveraging pretrained models like OpenCVlbp (Local Binary Patterns) is highlighted to enhance adaptability.

Target Audience and User Benefits: The online proctoring system is designed to benefit a diverse range of users, including students, professionals, and residents who rely on online examinations and learning courses. Discussions emphasize the importance of catering to the needs of these users by providing a dependable and secure examination environment.Functionality and Features: Key discussions focus on the functionality of the proctoring system, including its use of OpenCV for face identification and ensuring exam integrity. Features such as course practice, attending exams, and reports generation are highlighted to enhance user experience. The system's ability to prevent cheating by detecting tab switching, key pressing during exams is also emphasized.Implementation Considerations: Discussions address the technical aspects of implementing the proctoring exam system, highlighting the collaboration between OpenCV modules and deep learning concepts like CNN. Ensuring impersonation-free examinations through proctoring integration is emphasized.

Screenshots

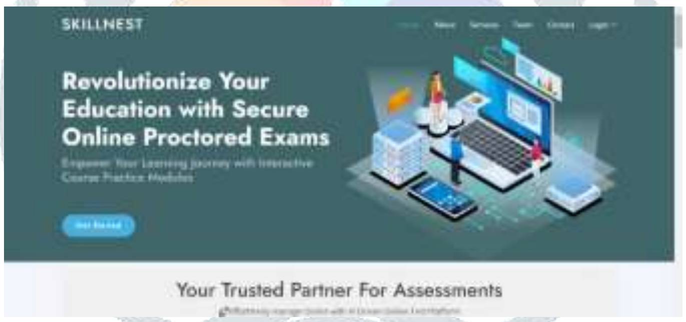

SKILLNEST
Revolutionize Your Education with Secure Online Proctored Exams
Your Trusted Partner For Assessments

Fig 1: Homepage

Imange-1 : This is the page where both the admin and student logins to their account via login pages. It describes about the project services and team along with contact page.

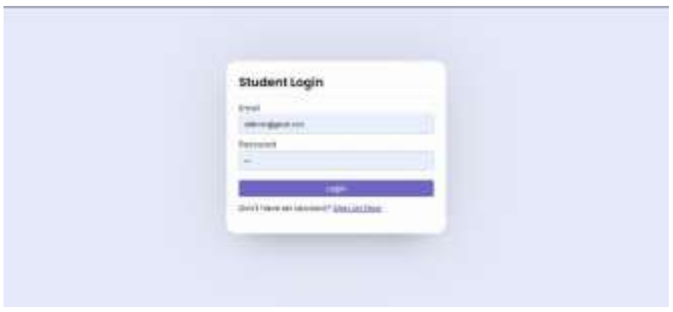


Fig 2: Student Login Page

Image-2: This is the page where a student logins to his/her registered account using his/her credentials like email, password.

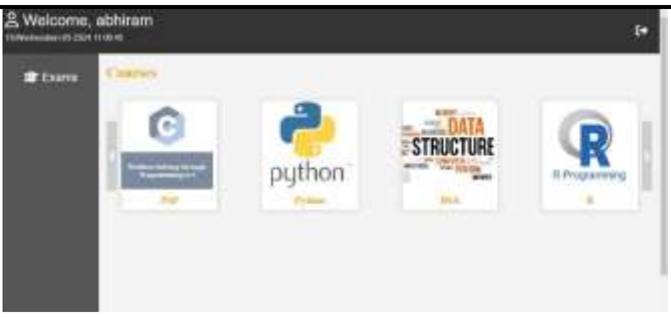
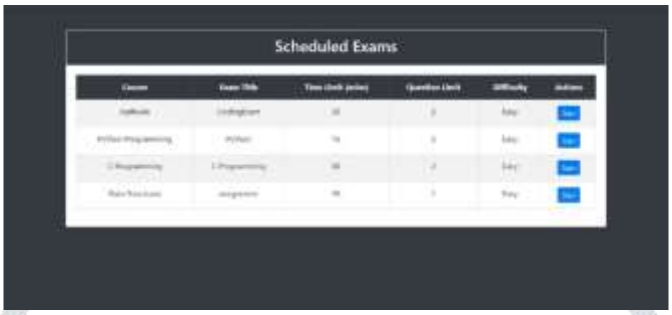


Fig 3: Student Dashboard

Image-3: The User interface of the student where he/she can able to do the courses and attend the exams is the student dashoard.



Scheduled Exams

Course	Exam Title	Time Limit (mins)	Question Limit	Difficulty	Actions
[illegible]	[illegible]	[illegible]	[illegible]	Easy	[illegible]
Python Programming	Python	[illegible]	[illegible]	Easy	[illegible]
C Programming	C Programming	[illegible]	[illegible]	Easy	[illegible]
Data Structures	assignment	[illegible]	[illegible]	Easy	[illegible]

Fig 4: Scheduled Exams of Student

Image-4: This Page consists of the scheduled exams for the student, where student enters to the exam and then marks his/her attendance.

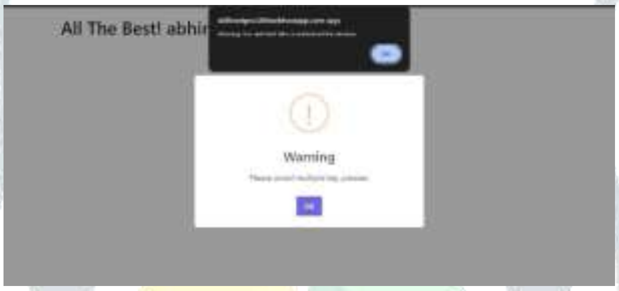


Fig 5: Face Capture Warning

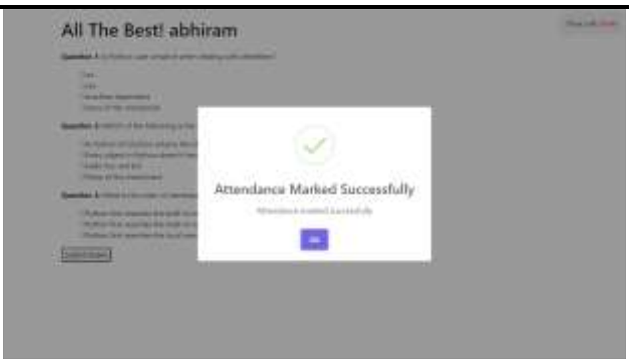
Image-5: Face Capture Warning

The system sends warning if any tab switches happened or face not recognized, by analayzing the face using OpenCV LBPH

Fig 6: Attendance Marking by Capturing Face

Image-6: Attendance Marking by Capturing Face

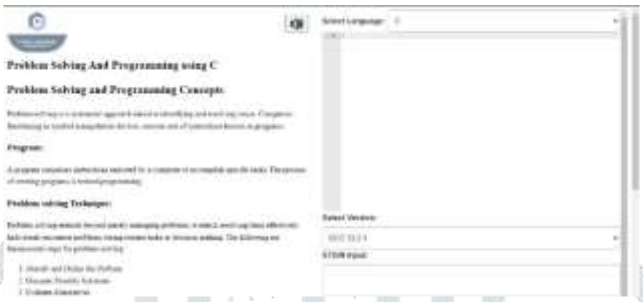
In the attendance marking page, Student has to mark his/her attendance after entering to the exam, By which the student attendance is noted in to database.


All The Best! abhiram

Attendance Marked Successfully

Fig 7: Successful Attendance

Image-7: Only after the successful attendance marking, the test paper will be opened, where student can attempt their exam.


Problem Solving And Programming using C

Problem Solving and Programming Concepts

Program:

Problem solving Techniques:

Fig 8: Integrated Compiler in Courses

Image-8: This integrated compiler is used to learn and practice the coding languages, where this compiler supports languages like C, C++, Python, Java,etc.

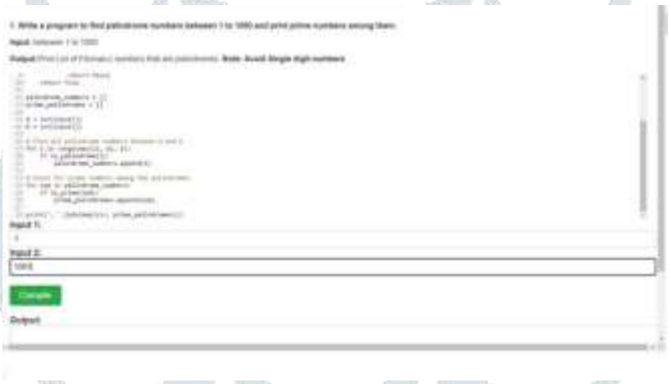


Fig 9: Exam Compiler

Image-9: This Exam Compiler is developed using the Nodejs compilex, which is the package/Extension used for developing compiler for running multiple programming languages.

Admin Login

Fig 10: Admin Loginpage

Image-10: This is the page where a Admin logins to his/her registered account using his/her credentials like email, password.

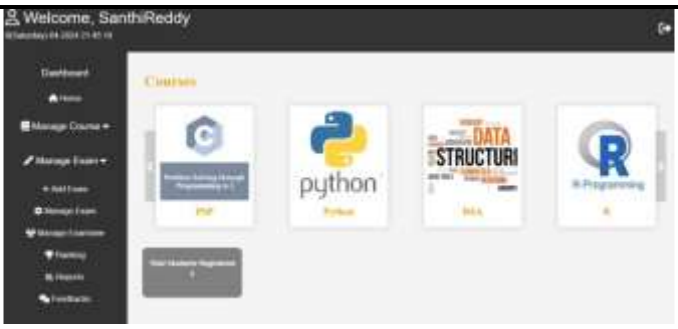


Fig 11: Admin Dashboard

Image-11: The User interface of the student where he/she can able to do their tasks like adding courses, exams, and view student reports is the Admin dashoard.

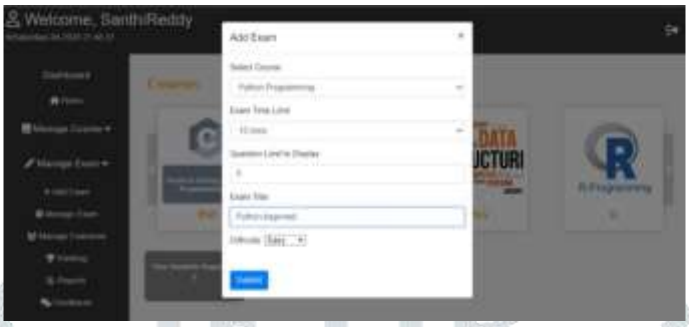


Fig 12: Adding exam

Image-12: Admin submits a form by mentioning the course title, course level, expected date for adding courses, phone number.

The standard deviations for each variable indicated that data were widely spread around their respective means.
Column 6 in table 4.1 shows jarque bera test which is used to checkthe normality of data. The hypotheses of the normal distribution are given;

H_0 : The data is normally distributed.
H_1 : The data is not normally distributed.
Table 4.1 shows that at 5 % level of confidence, the null hypothesis of normality cannot be rejected. KSE-100 index and macroeconomic variables inflation, exchange rate, oil prices and interest rate are normally distributed.
The descriptive statistics from Table 4.1 showed that the values were normally distributed about their mean and variance. This indicated that aggregate stock prices on the KSE and the macroeconomic factors, inflation rate, oil prices, exchange rate, and interest rate are all not too much sensitive to periodic changes and speculation. To interpret, this study found that an individual investor could not earn higher rate of profit from the KSE. Additionally, individual investors and corporations could not earn higher profits and interest rates from the economy and foreign companies could not earn considerably higher returns in terms of exchange rate. The investor could only earn a normal profit from KSE.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all those who have contributed to the completion of this research paper. First and foremost, we extend our heartfelt thanks to our Head of Department, Dr. C. Rajendra, whose insightful guidance and unwavering support have been invaluable throughout the course of this study at Narayana Engineering College.

We are also thankful to Narayana Engineering College for providing us with the necessary resources and facilities to conduct this research effectively.

Lastly, we acknowledge the participants who generously contributed their time and cooperation, without whom this study would not have been possible.

We are deeply appreciative of everyone's contributions and support.

REFERENCES

[1] Atoum, Yousef & Chen, Liping & Liu, Alex & Hsu, Stephen & Liu, Xiaoming. (2017). Automated Online Exam Proctoring. IEEE Transactions on Multimedia. PP. 1-1. 10.1109/TMM.2017.2656064.
[2] Li, Haotian & Xu, Min & Wang, Yong & Wei, Huan & Qu, Huamin. (2021). A Visual Analytics Approach to Facilitate the Proctoring of Online Exams.
[3] "Online Exam Proctoring System", A.T. Awaghade, D. A. Bombe, T. R. Deshmukh, K. D. Takawane, International Journal of Advance Engineering and Research Development (IJAERD) "E.T.C.W", January -2017, e-ISSN: 2348 - 4470, print-ISSN: 2348-6406.