# INDEX

# 1.ABSTRACT

CertifyTrack is a state-of-the-art platform tailored for managing AICTE activity points, event certifications, and student-mentor interactions within educational institutions. Traditional systems often lack integration, leading to inefficiencies, data silos, and administrative burdens. CertifyTrack overcomes these challenges by offering a unified, secure, and automated solution. Leveraging technologies such as Django and PostgreSQL, the platform ensures robust data management and scalability.

Core features include role-based access control for students, mentors, and clubs, streamlined event management, automated certificate generation, and mentor-allocated activity point validation. By reducing reliance on manual processes, CertifyTrack enhances accuracy, fosters collaboration, and aligns with institutional workflows. The system’s user-friendly design, coupled with secure data handling and scalability, makes it an indispensable tool for modern educational environments.

In the future, the platform aims to incorporate predictive analytics, real-time dashboards, and mobile applications, expanding its reach and functionality. CertifyTrack thus represents a significant advancement in the realm of educational process management, combining innovation with practical usability to create lasting value for institutions.

# 2.PROBLEM DEFINITION AND OBJECTIVES

## 2.1 Problem Definition

The rapid advancement of technology in the educational sector has highlighted the limitations of traditional systems for managing administrative tasks such as AICTE activity points, event certifications, and student-mentor interactions. Institutions often rely on disparate systems, spreadsheets, or paper-based processes, resulting in significant inefficiencies, data silos, and increased administrative burdens. These challenges hinder collaboration, delay critical workflows, and compromise the accuracy of record-keeping, ultimately impacting the overall educational experience.

1. Inefficient Management of AICTE Activity Points
    - Manual tracking of activity points leads to errors, inconsistencies, and delays.
    - Lack of a centralized system for managing activity points makes it difficult to monitor student progress effectively.
2. Disjointed Event Certification Processes
    - Generating and distributing certificates manually is time-consuming and prone to errors.
    - Absence of a secure and automated mechanism to ensure the authenticity of certifications.
3. Challenges in Student-Mentor Interactions
    - Ineffective communication channels between students and mentors hinder mentorship quality.
    - Lack of structured processes for allocating and validating mentor feedback on activities.
4. Data Silos and Integration Challenges
    - Fragmented systems lead to redundant data entry and inconsistencies across platforms.
    - Difficulty in integrating with existing institutional software, limiting scalability and adaptability.
    - Security vulnerabilities due to lack of robust data handling and role-based access control.

## 2.2 Objectives

CertifyTrack aims to address the above challenges by providing a comprehensive, unified platform tailored for the specific needs of educational institutions. The platform's objectives include:

1. **Automate AICTE Activity Points Tracking**
    - Streamline the recording and validation of activity points for accuracy and efficiency.
2. **Secure and Efficient Event Certification**
    - Simplify the generation and distribution of event certifications with secure processes.
3. **Structured Student-Mentor Communication**
    - Enhance mentorship through organized communication and feedback mechanisms.
4. **Seamless System Integration**
    - Integrate smoothly with existing institutional systems for unified workflows.
5. **Enhanced Data Security**
    - Ensure robust protection of sensitive data and adherence to compliance standards.
6. **Advanced Analytics and Mobile Access**
    - Provide actionable insights and on-the-go functionality through analytics and mobile apps.

# 3 LITERATURE REVIEW

## 3.1 Introduction

The need for a centralized and efficient system to manage institutional processes such as activity point tracking and certification has been recognized globally. Existing solutions often operate in silos, resulting in inefficiencies and user dissatisfaction. CertifyTrack addresses these challenges by offering an integrated platform with role-based functionalities.

## 3.2 Challenges in Existing Systems

1. **Fragmentation**: Most educational institutions rely on multiple, disconnected tools to manage events, certificates, and student-mentor interactions. This lack of integration leads to data redundancy, delays, and errors.
2. **Security Risks**: Without secure data storage and encryption, sensitive information such as student records and certificates is vulnerable to breaches. Existing platforms often fail to provide robust security measures.
3. **Scalability Issues**: Current systems often struggle to handle increasing user loads, making them unsuitable for large institutions or growing user bases.
4. **Manual Workload**: Many institutions still use manual processes for certificate generation and verification, increasing administrative overhead and the likelihood of errors.

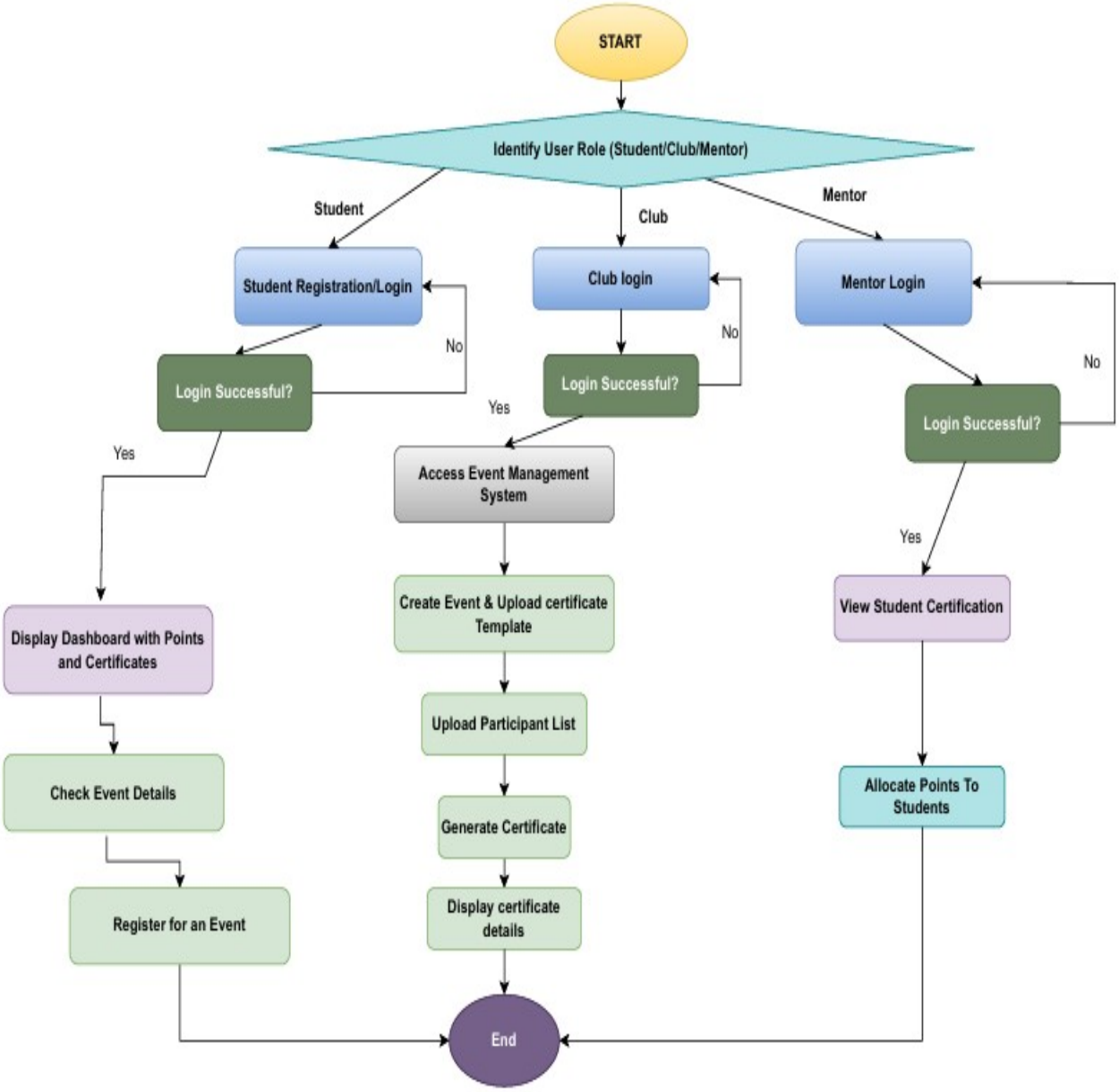
## 3.3 Existing Solutions and Gaps

1. **Certificate Management Tools:**While platforms like CertifyMe and Accredible focus on certificate management, they lack integration with event and AICTE activity point tracking. This creates additional dependencies on other tools.
2. **Learning Management Systems (LMS):**Popular LMS platforms such as Moodle and Blackboard include basic certification features but do not provide dedicated support for AICTE point tracking or mentor-student collaboration.
3. **Custom-Built** Solutions:Institutions with custom-built systems often face high maintenance costs and limited scalability, leaving them unable to adapt to evolving requirements.

## 3.4 Justification for CertifyTrack

1. CertifyTrack bridges these gaps by offering a unified platform that:
2. Integrates certificate management, event organization, and activity point tracking.
3. Provides robust security features, including role-based access control and encryption.
4. Ensures scalability to handle growing user bases and institutional needs.
5. Reduces administrative workload through automation and streamlined workflows.

# 4.DESIGN DETAILS



# 5. REQUIREMENTS ANALYSIS AND FEASIBILITY

## 5.1 Requirement analysis

### 5.1.1 Functional Requirements

1. Authentication: Secure login with role-based access for students, mentors, and clubs.
2. Profile Management: Allow users to update personal information and track AICTE points.
3. Event Management: Enable clubs to create and manage events with certificate templates.
4. Certificate Handling: Generate, store, and verify certificates.
5. Mentor-Student Allocation: Facilitate mentorship assignments for AICTE point validation.

### 5.1.2 Non-Functional Requirements

1. Performance: Ensure smooth operation for 100+ concurrent users.
2. Reliability: Maintain 99% uptime with robust data backup policies.
3. Security: Implement encryption for sensitive data.
4. Usability: Provide an intuitive, mobile-friendly user interface.

## 5.2 Feasibility Analysis

### 5.2.1 Technical Feasibility

- Utilizes Django and PostgreSQL for scalable, robust, and secure data management.
- Enables seamless integration with institutional systems through APIs.
- Incorporates advanced security measures like role-based access and encryption to safeguard data.

#### 5.2.2 Operational Feasibility

- Offers an intuitive user interface requiring minimal training for adoption.
- Automates manual processes, reducing administrative workload and errors.
- Aligns with institutional objectives by improving collaboration and workflow efficiency.

#### 5.2.2 Economic Feasibility

- Employs open-source technologies to minimize development and licensing expenses.
- Reduces long-term operational costs through process automation.

# 6.FUNCTIONAL/NON-FUNCTIONAL REQUIREMENTS

## 6.1 Functional Requirements

### 6.1.1. User Role Management

- **Role-Based Access Control (RBAC):**
  - Students: Access activity points, submit proof for activities, view certifications, and check progress.
  - Mentors: Validate activity points submitted by students, review certification processes, and provide feedback.
  - Clubs/Administrators: Manage events, monitor student participation, and handle certificate generation.
- Authentication and Authorization:
  - Secure login for all users with appropriate access privileges.
  - Support for password reset and user profile management.

### 6.1.2. Activity Point Management

- Submission of activity proofs by students (e.g., document uploads or event details).
- Mentor review and validation of activity submissions.
- Automatic calculation of accumulated activity points for students.
- Notifications to students and mentors about pending validations or rejected submissions.

### 6.1.3. Event Management

- Creation and scheduling of events by administrators or clubs.
- Participant registration with an integrated interface.
- Real-time event updates and notifications to registered users.
- Tracking attendance and participation.

### 6.1.4. Certificate Generation and Management

- Automated generation of certificates for event participants.
- Storage of digital copies of certificates for student access.
- Customizable certificate templates based on event types or institutional standards.

### 6.1.5. Data Reports and Analytics

- Generate reports on student activity points, event participation, and certifications.

- Overview dashboards for mentors and administrators to monitor engagement and progress.

#### 6.1.6. Communication and Notifications

- System-generated notifications for pending actions, validations, and updates.
- In-app messaging for student-mentor communication.

#### 6.1.7. Integration with Institutional Systems

- APIs to integrate CertifyTrack with existing Learning Management Systems (LMS) or other academic platforms.
- Support for exporting data in standard formats (e.g., CSV, Excel).

## 6.2 Non-Functional Requirements

These outline the operational and quality attributes of the system.

#### 6.2.1. Performance

- The platform should handle concurrent access by hundreds of users without significant latency.
- Ensure quick response times (<2 seconds for most actions like form submissions or page transitions).

#### 6.2.2. Scalability

- The system must scale to accommodate increasing numbers of users, events, and data.
- Flexible database schema (PostgreSQL) to handle growth in stored activity records and certificates.

#### 6.2.3. Security

- Use Django's built-in security features for secure authentication (e.g., hashed passwords).
- Role-based access to restrict unauthorized data viewing or actions.
- Regular backups of critical data to prevent loss.

#### 6.2.4. Reliability

- Ensure system uptime of 99.9% or higher.
- Implement automated testing to reduce bugs and ensure smooth updates.

#### 6.2.5. Usability

- Design an intuitive and user-friendly interface for all user roles.
- Provide comprehensive user documentation and support for onboarding.

#### 6.2.6. Data Integrity

- Ensure accurate recording and validation of activity points and certifications.
- Prevent duplication or loss of data through transaction rollbacks and integrity constraints in the database.

#### 6.2.7. Maintainability

- Modular codebase using Django to facilitate easy updates and bug fixes.
- Proper documentation of the codebase for future developers.

### 6.2.8. Compliance

- Align with institutional and governmental regulations (e.g., AICTE guidelines) for activity points and certifications.
- Ensure adherence to data protection laws, such as GDPR, for student and institutional data.

### 6.2.9. Portability

- Cross-browser compatibility for popular browsers (e.g., Chrome, Firefox, Edge).
- Platform-independent design to support deployment on various server environments.

### 6.2.10. Future Expansion

- Incorporate predictive analytics for student performance and event success metrics.
- Develop mobile applications for on-the-go access.

# 7. CODE

```python
@login_required
def view_certificate(request, certificate_id):
    try:
        certificate = Certificate.objects.get(id=certificate_id)

        # Ensure the user is authorized to view the certificate
        if (
            request.user.profile.role == 'student' and certificate.participant.student != request.user or
            request.user.profile.role == 'mentor' and certificate.participant.student.profile.mentor != request.user
        ):
            return HttpResponse("Unauthorized", status=403)

        # Get the file path for the certificate
        certificate_path = certificate.certificate_file.path

        # Read the file from the file system
        with open(certificate_path, 'rb') as cert_file:
            file_data = cert_file.read()

        # Determine content type
        if file_data.startswith(b'%PDF'):
            content_type = 'application/pdf'
        elif file_data[:3] == b'\xff\xd8\xff':  # JPEG marker
            content_type = 'image/jpeg'
        elif file_data[:8] == b'\x89PNG\r\n\x1a\n':  # PNG marker
            content_type = 'image/png'
        else:
            raise Http404("Unsupported file format.")

        # Serve the file
        response = HttpResponse(file_data, content_type=content_type)
        response['Content-Disposition'] = f'inline; filename="certificate_{certificate.id}.{content_type.split("/")[-1]}"'
        return response

    except Certificate.DoesNotExist:
        raise Http404("Certificate not found.")
```

**Figure 7.1**

```python
def generate_event_certificates(request, event_id):
    try:
        for participant in participants:
            try:
                # Debug: Ensure participant is an instance of Participant
                print(f"Processing participant: {participant} (ID: {participant.id})")

                # Get participant details
                student_profile = participant.student.profile  # Get the student's profile
                full_name = student_profile.full_name
                usn = student_profile.usn

                # Debug: Check participant details
                print(f"Participant Name: {full_name}, USN: {usn}")

                # Prepare paths for template and output
                output_path = os.path.join(temp_dir, f"certificate_{participant.id}.{file_type}")

                # Handle PDF templates
                if file_type == "pdf":
                    # Read the existing PDF template
                    with open(template_path, "rb") as pdf_file:
                        pdf_reader = PdfReader(pdf_file)
                        pdf_writer = PdfWriter()

                        # Go through each page of the PDF template
                        for page in pdf_reader.pages:
                            packet = BytesIO()
                            can = canvas.Canvas(packet, pagesize=letter)

                            # Add participant details on top of the existing template
                            can.setFont("Helvetica-Bold", 14)
                            can.drawString(200, 300, f"Name: {full_name}")
                            if usn:
                                can.drawString(200, 270, f"USN: {usn}")
                            can.save()
```

**Figure 7.2**

# 8. SYSTEM DESIGN

## 8.1 Architectural Design

1. Frontend:
    - Developed using HTML, CSS, and JavaScript, ensuring a responsive and intuitive user interface.
    - Utilizes frontend frameworks like Bootstrap for styling and responsiveness.
    - Implements AJAX for dynamic content updates and seamless user interactions.
2. Backend:
    - Powered by the Django Framework, which provides a robust backend with RESTful API capabilities.
    - Follows the Model-View-Template (MVT) architecture to facilitate separation of concerns.
    - Incorporates Django's built-in security features, such as CSRF protection and authentication.
3. Database:
    - PostgreSQL serves as the relational database management system, ensuring data integrity and scalability.
    - Utilizes advanced indexing, foreign key constraints, and normalization for efficient data retrieval and storage.
    - Regular backups and database replication strategies ensure data availability and disaster recovery.
4. Hosting:
    - The application is deployed on AWS/Heroku, ensuring scalability and high availability.
    - Leverages AWS services such as EC2 for hosting, RDS for database management, and S3 for storing certificates and related assets.
    - Implements CI/CD pipelines for automated deployment and updates.

## 8.2 Component Design

1. Authentication:
    - Implements a role-based access control (RBAC) system to define permissions for students, mentors, and clubs.
    - Utilizes Django's built-in authentication system with additional support for OAuth and social login options.
    - Features multi-factor authentication (MFA) for enhanced security.

2. Event Management:
    - Provides a complete CRUD (Create, Read, Update, Delete) interface for managing events.
    - Allows users to upload certificate templates, which are automatically populated with event and participant details.
    - Incorporates event scheduling, notification system, and analytics to track participation.
3. Mentor Allocation:
    - Tracks mentorship relationships between students and mentors.
    - Facilitates mentor-assigned activity point validation with automated tracking.
    - Provides reporting tools for mentors to evaluate student progress and contributions.

## 8.3 Future Enhancements

- Predictive Analytics: Leverage AI/ML to provide insights into student engagement and performance.
- Real-time Dashboards: Interactive dashboards for monitoring event participation and activity point trends.
- Mobile Applications: Native apps for Android and iOS to enhance accessibility and usability.

# 9. PROPOSED VIABLE SOLUTION

CertifyTrack is designed as a comprehensive solution to the challenges of managing AICTE activity points, event organization, and certification workflows. By integrating multiple functionalities into a single platform, the system eliminates redundancies, improves collaboration, and ensures data security. Below are the key aspects of the proposed solution:

1) **Centralized AICTE Point Management**:
    i) CertifyTrack provides a unified database to track and store students' activity points. This reduces the need for manual calculations and ensures transparency in AICTE compliance.
2) **Efficient Event Management**:
    i) Clubs can create, edit, and manage events with ease. Features include linking events to certificate templates, managing participants, and tracking event statuses (e.g., Not Started, Started, Finished).
3) **Seamless Certificate Handling**:
    i) Certificates are generated automatically based on uploaded templates, ensuring accuracy and reducing administrative workload. The system securely stores certificates and allows easy retrieval for verification.
4) **Enhanced Collaboration**:
    i) The platform facilitates interaction between students, mentors, and club administrators. Mentors are assigned specific students to verify their activity points, streamlining the verification process.
5) **Role-Based Access Control:**
    i) CertifyTrack uses a robust authentication mechanism that provides different levels of access based on the user's role (Student, Mentor, Club). This ensures that users have access only to the functionalities relevant to them.
6) **Scalability and Modularity:**
    i) Built on Django and PostgreSQL, the system is designed to handle increasing user loads and adapt to institutional growth. Its modular architecture allows for the addition of new features with minimal disruption.

# 10. Conclusion

CertifyTrack effectively addresses inefficiencies in activity point tracking and certificate management by providing a centralized, secure, and scalable platform. It fosters collaboration among stakeholders, ensuring streamlined operations and compliance with AICTE requirements. By reducing manual processes and integrating key functionalities such as role-based access control, automated certificate generation, and mentor validation, CertifyTrack enhances operational efficiency and accuracy. With future advancements in predictive analytics and mobile accessibility, CertifyTrack aims to remain a cutting-edge solution in educational process management, delivering lasting value to institutions and their stakeholders.

.

## Future Scope

1. eliminate need for physical certificate
2. reduce work of mentor
3. transform this into an event management system
4. Permission , venue booking , Announcement etc can be centralized