



AN IMS [ISO 21001, ISO 9001, ISO 14001, ISO 45001] CERTIFIED INSTITUTION

NETTUR TECHNICAL TRAINING FOUNDATION

PROJECT REPORT ON PRESENT VIA PIXEL

DIPLOMA IN INFORMATION TECHNOLOGY & DATA SCIENCE 5TH
SEMESTER

PROJECT WORK DONE BY:

NEC0923041- GUNJAN

**DIPLOMA IN
INFORMATION TECHNOLOGY AND DATA SCIENCE
NTTF ELECTRONICS CENTRE
ELECTRONIC CITY BANGALORE -560100**

2023 - 2026



AN IMS [ISO 21001, ISO 9001, ISO 14001, ISO 45001] CERTIFIED INSTITUTION

NETTUR TECHNICAL TRAINING FOUNDATION

CERTIFICATE

This is to certify that the project titled

PRESENT VIA PIXEL

Is a Bonafide report of report work done by

NEC0923041- GUNJAN

IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DIPLOMA IN INFORMATION TECHNOLOGY AND DATA SCIENCE(CP09) UNDER THE INSTITUTION NETTUR TECHNICAL TRAINING FOUNDATION, ELECTRONIC, CENTER, BANGALORE DURING THE ACADEMIC YEAR 2023-2026.

PROJECT GUIDE

PROJECT COORDINATOR

EXTERNAL EXAMINER

COURSE HEAD

ACKNOWLEDGEMENT

I would like to express my deepest gratitude and appreciation to all those who have supported me in the completion of this final year project.

To begin with, I would like to sincerely thank my Principal, **Mr. Somanathan K (NEC-NTTF)**, for his constant encouragement and support which motivated me to do my best. I am also thankful to my Course Head and Program Manager of CP09, **Mr. Jagadish G. K**, for being a continuous source of inspiration throughout the project.

I extend my heartfelt thanks to my **Project Coordinator, Mrs. Bhuvana K**, for her invaluable guidance, constant encouragement, and constructive feedback. Her expertise and mentorship have played a crucial role in shaping this project and enhancing its quality. I am especially indebted and deeply grateful to my **Project Guide, Mrs. Bhuvana K**, for her valuable guidance and support at every stage of this project, which helped turn my ideas into reality.

In conclusion, this project would not have been possible without the collective efforts and support of all those mentioned above and my team members. I am truly grateful for their contributions and feel fortunate to have been surrounded by such incredible individuals during this journey.

ABSTRACT

The **Present via Pixel** is an automated solution designed to record attendance using facial recognition and time-based validation. The system captures live video through a webcam, detects faces, and compares them with the pre-registered facial encodings stored in the system. When a match is found, the attendance is marked automatically with the exact date and time.

This project is developed using **Python**, **OpenCV**, the **face_recognition** library, and the **Flask** framework to provide a simple and user-friendly web interface. The system includes essential modules such as student registration, face image encoding, live attendance monitoring, and an attendance dashboard.

A key feature of this project is **area-based validation**, where attendance is recorded only if the student is within the allowed camera radius. The system also detects **late entry**, marking students as “Late” if they arrive after a predefined cutoff time. This makes the attendance process more accurate and stricter compared to manual methods.

The main goal of this project is to eliminate manual entry errors, reduce proxy attendance, and provide a fast, contactless, and efficient attendance system suitable for educational environments. The modular design also allows easy updates and future improvements.

TABLE OF CONTENTS

SL NO.	CHAPTER NO.	CONTENTS	PAGE NO.
1	1	INTRODUCTION	6
2	1.1	AIM	6
3	1.2	OBJECTIVES	6
4	1.3	DEMERITS OF EXISTING SYSTEM	7
5	1.4	MERITS OF PROPOSED SYSTEM	8
6	2	SYSTEM ANALYSIS	11
7	2.1	FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS	11
8	2.1.1	FUNCTIONAL REQUIREMENTS	13
9	2.1.2	NON-FUNCTIONAL REQUIREMENTS	13
10	2.2	SYSTEM SPECIFICATIONS	13
11	2.2.1	HARDWARE SPECIFICATIONS	13
12	2.2.2	SOFTWARE SPECIFICATIONS	14
13	3	DESIGN AND DEVELOPMENT PROCESS	17
14	3.1	MODULES AND FUNCTIONALITIES	17
15	4	DESIGN METHODS	20
16	4.1	DATA FLOW DIAGRAM	20
17	4.2	USE CASE DIAGRAM	21
18	4.3	GANTT CHART	22
19	5	TESTING	23
20	6	IMPLEMENTATION	25
21	7	RESULTS	32
22	8	CONCLUSION	36

CHAPTER 1

INTRODUCTION

1.1 AIM

The aim of this project is to develop an **automated Present via Pixel** that accurately records student attendance using live facial detection and verification. The system intends to eliminate manual attendance errors, prevent proxy attendance, and ensure that attendance is marked only when the student is physically present within the allowed area radius. The project also aims to detect **late entries** by comparing the captured time with a predefined schedule. Overall, the system provides a fast, contactless, and reliable attendance solution using Python, OpenCV, face recognition, and a web-based interface built with Flask.

1.2 OBJECTIVES

- Automate Attendance Recording
- Enhance Security and Prevent Proxy Attendance
- Improve Efficiency and Save Time
- Provide Real-time Data and Reporting
- Offer a Touchless and Hygienic Solution

1.3 DEMERITS OF EXISTING SYSTEM

Traditional attendance systems have several operational and security weaknesses that limit accuracy, reliability, and transparency.

1. Manual Roll Calls Are Time-consuming and Inaccurate:

Manual attendance takes significant classroom time and often results in spelling mistakes, missed entries, and incorrect markings. Human dependency makes the process slow and unreliable—especially in large batches.

2. High Chances of Proxy Attendance:

Conventional systems allow students to respond on behalf of absent classmates. There is no strong identity verification, making it easy for proxies and manipulation.

3. Card/Biometric Devices Require Dedicated Hardware:

Systems like RFID or fingerprint scanners require costly hardware that needs regular maintenance. These devices suffer from:

- Card sharing and card loss
- Sensor malfunction
- Wear and tear due to continuous use

This increases operational cost and reduces long-term efficiency.

4. Camera Systems Without Security Checks Are Easily Spoofed:

Basic camera-based systems that lack liveness detection or geolocation validation can be fooled using printed photos, mobile displays, or recorded videos—leading to false attendance.

5. No Real-Time Digital Integration:

Traditional systems fail to generate automatic logs or export-ready attendance reports. Instructors still rely on manual Excel entries, which increases workload and error probability.

6. No Remote Access or Multi-Device Synchronization:

Attendance data stored locally cannot be accessed easily across departments or devices. Administrators cannot track records centrally or manage multiple classes from a unified system.

7. Low Scalability with Larger Student Groups:

As student count increases, manual or device-dependent systems become slow, inefficient, and difficult to manage.

8. Lack of Transparency and Verification:

Students cannot verify their attendance status, and teachers cannot review historical logs easily. This leads to disputes and a lack of accountability in record-keeping.

1.4 MERITS OF THE PROPOSED SYSTEM

The “**Present Via Pixel – Face Recognition Attendance System**” introduces a modern, secure, and automated approach to attendance tracking using facial recognition and geolocation validation.

1. Fast and Fully Contactless Attendance:

The system identifies students instantly through a live camera feed. No physical interaction—such as touching scanners or carrying cards—is required, making the process quick and hygienic.

2. Cost-Effective Implementation:

It runs on any general-purpose computer and a standard webcam. No special biometric device or paid software is needed. The use of open-source libraries reduces overall cost.

3. Enhanced Security Through Geofence Verification:

The system validates whether the user is within the approved location radius, preventing students from marking attendance from home or outside the institution.

4. Consecutive Frame Validation for High Accuracy:

The system only marks attendance after confirming the face match in multiple consecutive frames. This minimizes false positives caused by lighting, angle changes, or sudden motion.

5. Role-Based Access for Better Control:

Different users get different levels of access:

- **Admin:** Manage students, records, and system settings
- **Teachers:** View and monitor class attendance
- **Students:** Mark attendance and check their own attendance
This ensures security and better data organization.

6. Automatic and Well-Structured Data Logging:

Each attendance record is stored with:

- Token Number
- Name
- Date & Time
- Late/On-Time status

Reports can be exported easily in **CSV** or **Excel (XLSX)** format.

7. Modular, Scalable, and Easy to Upgrade:

The system is developed with a modular structure, making it easy to add new features such as:

- Notifications
- Detailed analytics
- Department-wise dashboards
It also works efficiently even when the student database grows.

8. High Accuracy Through Advanced Facial Encoding:

The recognition engine uses 128-dimensional face encodings, resulting in precise identification even for students with similar facial features.

9. Clean, Responsive, and User-Friendly Web Interface:

Built using Flask and Bootstrap, the interface is easy to navigate and works smoothly across laptops and desktops. Live camera streaming and status notifications improve the overall experience.

10. Strong Data Security and Privacy Protection:

All images and encodings are stored locally. No data is uploaded to third-party servers, ensuring confidentiality and reducing privacy risks.

11. Simple Installation and Platform Flexibility:

Requires only Python, pip, and a webcam. Works on Windows and Linux systems without complex configuration.

12. Automatic Late Entry Detection:

The system checks the timestamp against predefined class timings and labels students as **On-Time** or **Late**, enabling better discipline and record management.

CHAPTER 2

SYSTEM ANALYSIS

2.1 FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

2.1.1 Functional Requirements

1. User Authentication:

- The system must allow secure login and logout.
- Only authorized users (Admin/Teacher/Student) can access dashboards.
- Incorrect login attempts must be rejected with proper error messages.

2. Student Data Management:

- The system must allow adding, editing, and deleting student records.
- The system must store each student's basic details (token no, name, course, batch).
- The system must validate duplicate entries before saving.

3. Face Data Enrolment:

- The system must allow uploading a student's face image.
- It must generate accurate face encodings for each registered student.
- The system must store encoding files in a structured format.
- Re-encoding must be possible when required.

4. Liveness Verification:

- The system must ensure that the person in front of the camera is real and not a spoof.
- It must detect blinking or natural head movement.
- Attendance must be marked only after liveness confirmation.

5. Real-Time Face Recognition:

- The system must detect faces from the live camera stream.
- It must match the detected face with pre-stored encodings.
- The system must confirm identity after N consecutive matches.
- Unknown faces must be ignored or flagged.

6. Attendance Marking:

- The system must record:
 - Student name
 - Token number
 - Date
 - Time
 - Attendance status (On-Time / Late)
- Duplicate attendance for the same day must be prevented.
- Logs must be stored automatically.

7. Geofence Validation:

- The system must verify the user's location before allowing attendance.
- If the user is outside the allowed radius, attendance must be blocked.

8. Attendance Logs & Reporting:

- The system must store attendance data in a structured format.
- It must allow viewing attendance day-wise and student-wise.
- The system must support exporting attendance in CSV and Excel formats.

9. Live Camera Streaming:

- The system must provide an accessible /video feed endpoint.
- The camera stream must load smoothly in the browser.
- The feed must stop automatically when the process ends.

10. Multi-Role Dashboard:

- Admin dashboard must allow:
 - Student management
 - Encoding reset
 - Attendance overview
 - Export
- Student/Teacher dashboard must display relevant attendance only.

2.1.2 Non-Functional Requirements

- **Performance:** Stream at reasonable FPS (dependent on hardware), face detection at reduced resolution for speed.
- **Reliability:** Avoid duplicate inserts, handle camera disconnects gracefully.
- **Scalability:** Support increasing number of registered students (encoding loading strategy).
- **Usability:** Simple, responsive UI.
- **Security:** Password hashing for default users, restrict endpoints to logged-in users.

2.2 SYSTEM SPECIFICATIONS

2.2.1 Hardware Specifications:

- **Processor:** Intel core i3 or equivalent
- **RAM:** 4 GB or more
- **Storage:** 10–15 GB free disk space
- **Camera:** Standard USB Webcam (720p resolution)
- **Network:** Basic internet/Wi-Fi for geolocation services

2.2.2 Software Specifications:

1. Front-End:

The front-end of the system is designed to provide a clean, user-friendly interface for students and administrators.

Technologies Used:

- **HTML5** – Structure of all web pages
- **CSS** – Styling, layout, themes, gradients, animations
- **JavaScript** – Basic validation and UI interactivity

Purpose:

Ensures a smooth and responsive user interface for dashboards, student management, and attendance viewing.

2. Back-End:

The backend handles all business logic, camera processing, face recognition, and database coordination.

Programming Language

- **Python 3.9** – Core programming language for processing, camera access, and server logic.

Required Python Libraries

- **face_recognition** – Generates 128-dimension facial encodings
- **OpenCV-python (OpenCV)** – Reads webcam video frames in real time
- **NumPy** – Numerical operations for encoding comparisons
- **Flask** – Backend framework for routing, API endpoints, and MJPEG streaming
- **Pillow (PIL)** – Image handling during registration
- **geopy** – Geofencing and radius-based location verification
- **pickle** – Saving and loading facial encoding files (.pkl)
- **pandas, openpyxl** – Exporting attendance reports in CSV and Excel formats

Framework

- **Flask Web Framework**

Used for:

- Routing and navigation
- Handling registration and login requests
- Live camera streaming (/video_feed)
- Serving dashboard pages
- Managing student data

Purpose:

Controls all server-side operations, security, data flow, and real-time recognition pipeline.

3. Database:

Database Management System

- **SQLite** – Lightweight, serverless, file-based DBMS

Database Structure

- Stores:

- Student details (token no, name, photo path, encoding path)
- Attendance logs (date, time, status)

DB File:

- A single .db file stores all data, simplifying deployment.

4. Development Tools:

Python IDE

- **PyCharm / VS Code**

Used for coding, debugging, and managing project structure.

Browser

- **Microsoft Edge / Google Chrome**

Used for running the frontend interface and viewing the live camera feed.

Server

- **Flask Development Server (localhost)**

- Runs all backend APIs
- Handles streaming and dashboard loading
- Suitable for local college-level deployment

CHAPTER 3

MODULES AND FUNCTIONALITIES

1. Student Registration Module

- Allows new student registration through a form-based interface.
- Supports photo upload directly from the browser.
- Automatically generates a unique face encoding for each registered student.
- Stores each encoding file separately (e.g., `student_id.pkl`).
- Validates duplicate entries to prevent multiple registrations for the same student.

2. Face Encoding & Storage Module

- Captures multiple face samples to improve recognition accuracy.
- Preprocesses images (resize, RGB conversion).
- Generates 128-dimensional facial feature vectors using the `face_recognition` library.
- Stores encoding files inside the `known_faces/` directory.
- Supports regeneration/reset of face encodings by admin.

3. Live Camera Stream Module

- Provides a browser-accessible live camera stream via `/video_feed`.
- Streams video in MJPEG format for fast rendering on web browsers.
- Activates camera only after geofence validation (if enabled).
- Supports smooth camera start/stop operations.

4. Face Detection & Recognition Module

- Detects one or more faces in real-time video frames.
- For each detected face:
 - Extracts real-time face encodings
 - Compares with known encodings
 - Computes similarity distances
 - Confirms identity if below threshold

- Requires **N consecutive matching frames** to finalize recognition.
- Ignores unknown faces or logs them for review.

5. Liveness Detection Module (Anti-Spoofing)

- Prevents spoofing attacks such as photo or video replay.
- Checks for natural blinking patterns.
- Detects head movements across frames to ensure a living subject.
- Optionally supports advanced liveness methods like:
 - Texture analysis
 - Depth/IR feedback (if hardware available)
- Only after liveness confirmation does the system allow attendance marking.
(This significantly increases the security and reliability of recognition.)

6. Attendance Capture Module

- Records attendance when identity and liveness are successfully verified.
- Stores:
 - Token number
 - Student name
 - Date
 - Time
 - Attendance status
- Prevents duplicate entries on the same day.
- Saves attendance logs inside attendance_data/.

7. Geofence Verification Module

- Validates the user's geographic location before attendance.
- Restricts attendance marking to predefined campus boundaries.
- Deactivates the camera module if geolocation is invalid.

8. Authentication & Role-Based Access

- Provides secure login/logout functionality.
- Supports two roles:
 - **Admin**
 - **User (Student/Teacher)**
- Displays role-specific dashboards:
 - Admin → Student management, analytics, export
 - User → View attendance only

9. Attendance Export Module

- Allows export of attendance records in:
 - CSV format
 - Excel (.xlsx) format
- Supports filter-based export (date-wise, student-wise, batch-wise).
- Ensures accurate formatting during export.

10. Admin Management Module

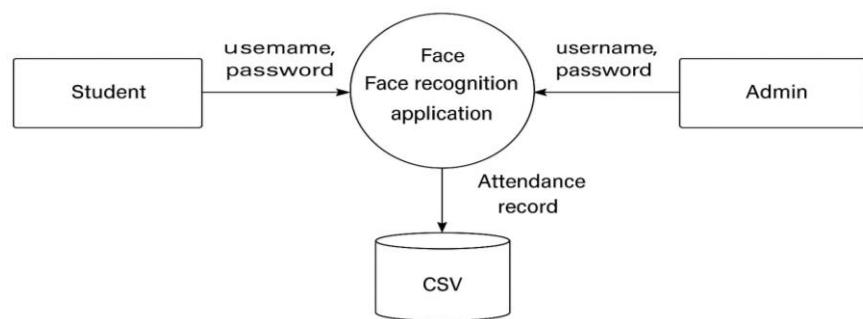
- Admin can add new students.
- Admin can edit or update student details.
- Admin can delete student records.
- Admin can reset or regenerate face encodings when necessary.
- All changes update instantly across all modules.

CHAPTER 4

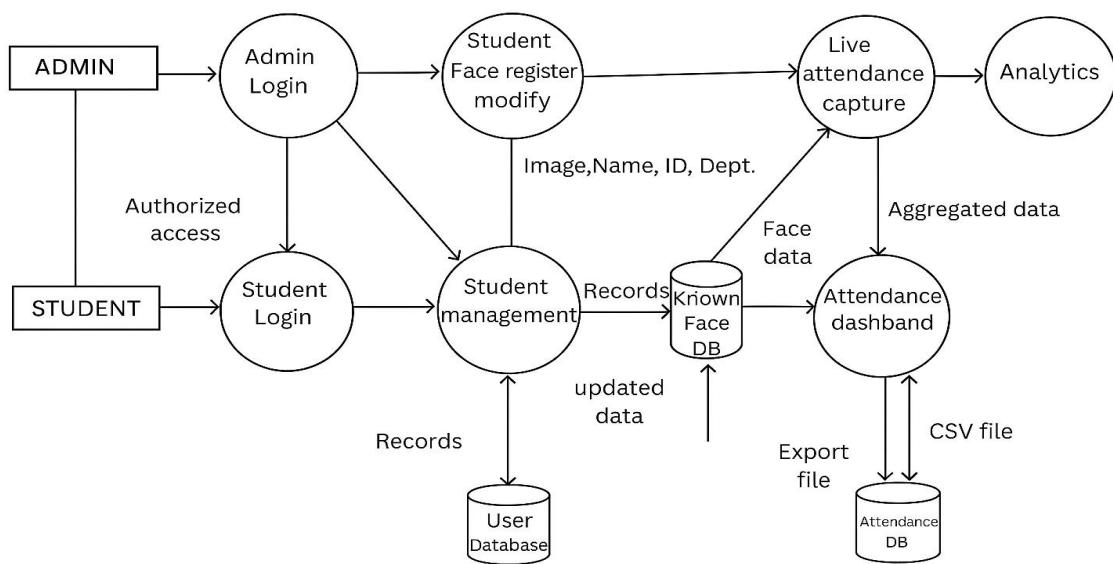
DESIGN METHODS

4.1 DATA FLOW DIAGRAM:

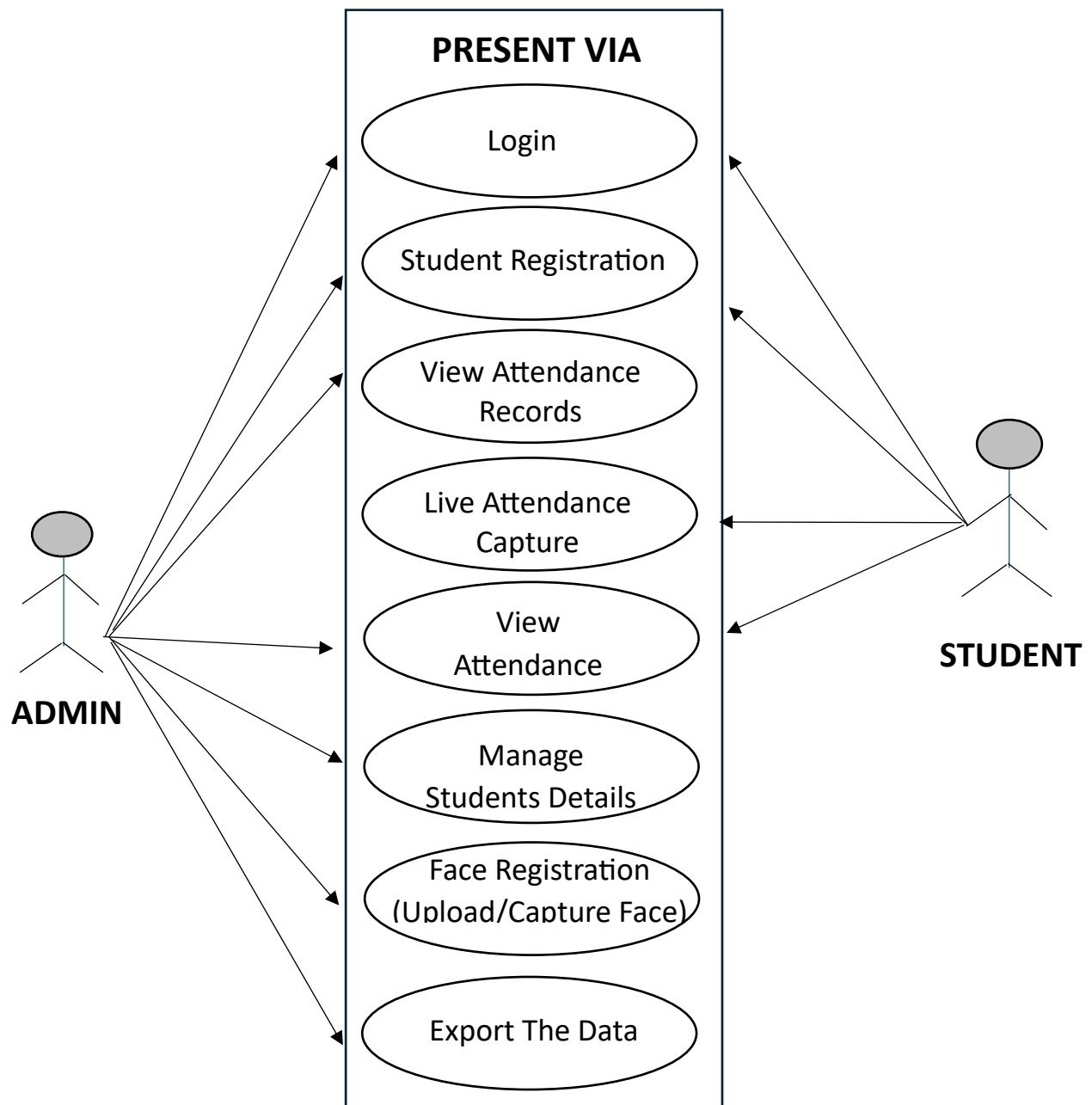
LEVEL 0 DFD



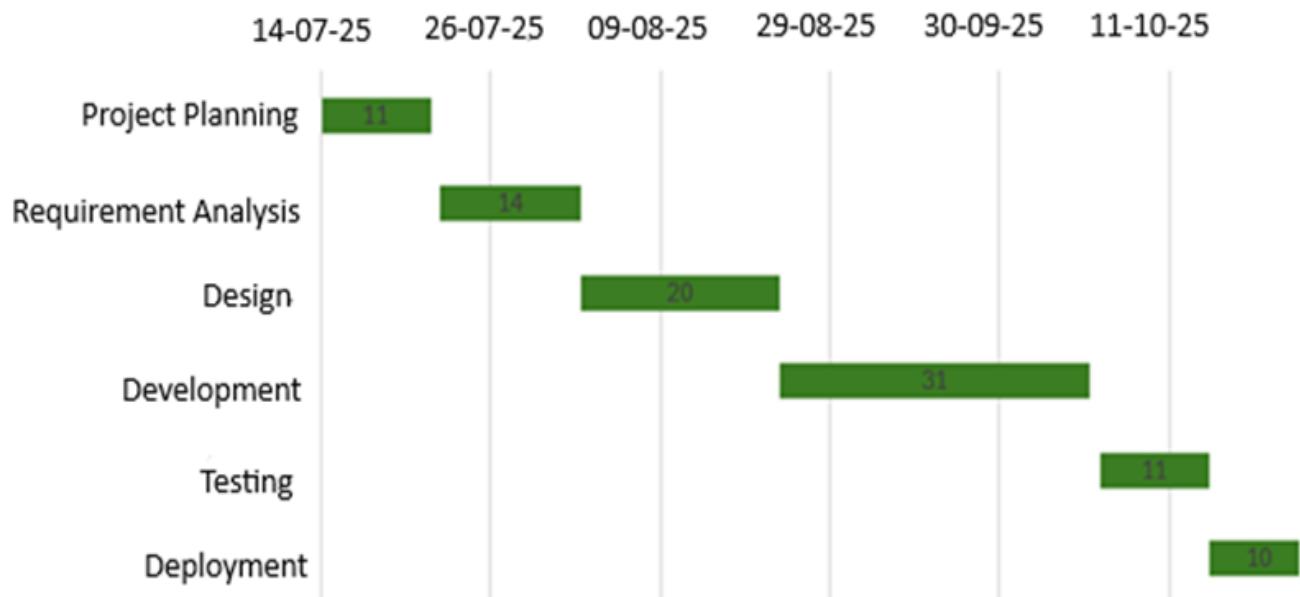
LEVEL 1 DFD



4.2 USE CASE DIAGRAM:



4.3 GANTT CHART:



The project timeline is planned in six phases, starting from July to October 2025. It begins with Project Planning, followed by Requirement Analysis and Design. Once the design is completed, the Development phase takes the most time as the main system is built. After development, the Testing phase ensures all features work correctly. Finally, the project ends with the Deployment phase, where the system is installed and made ready for use. This schedule ensures smooth progress and timely completion of the project.

CHAPTER 5

TESTING

TEST CASE:

SL NO	TC ID	Test Scenario	Test Description	Input / Steps	Expected Result	Actual Result	status
1	FR_01	Student new registration	Open “New Registration” page → username: Gunjan, password: Gun@123, role: student, click Register	Success message shown, student user created in users table with role=student, same credentials work on login	Success message displayed, Gunjan created in DB, login works with Gunjan/Gun@123	Working successfully	pass
2	FR-02	Face Recognition	Match live face with stored encodings	Student face	Correct student name appears after training	Works after training	pass
3	FR-03	Attendance Marking	Store attendance with timestamp	Recognized face	Record inserted in DB with Date, Time, Status	Correct entry added	pass
4	FR-04	Late Marking Logic	Check late detection	Mark attendance after allowed time	Status stored as “Late”	Status logged properly	pass
5	FR-05	Geofence Verification	Validate allowed campus area	GPS outside allowed radius	Message: “Access Denied – Out of Campus”	Validation successful	pass
6	FR-06	Admin Login	Validate admin credentials	Username + password	Redirect to admin dashboard	Works as expected	pass

7	FR-07	Add Student (Admin)	Register new student with token, name, face capture	Token: NEC09230 41, Name: Gunjan, 5-10 face frames	Student stored in DB and face encoding generated	Works as expected	pass
8	FR-08	Edit Student (Admin)	Edit existing student details	Update token or name	Database updated successfully	Works	pass
9	FR-09	Delete Student (Admin)	Remove student and encoding	Click delete on student record	DB record removed and related deleted	Works	pass
10	FR-10	Export CSV, Excel	Export attendance as CSV, Excel	Click "Export CSV or Excel"	Attendance CSV, XLSX file downloaded	Works	pass
11	FR-11	Multiple Faces	Handle two or more faces in same frame	Two or more faces visible	Attendance recorded;	Works correctly	pass
12	FR-12	Invalid Encoding	Handle corrupt or missing encoding file	Use unreadable or missing.pkl	Error message "Encoding file invalid" shown/handled	Error handled	pass
13	FR-13	Database Integrity	Ensure no duplicate attendance entries	Multiple scans for same student, same day	Only one entry per student per date in DB	DB clean	pass

CHAPTER 6

IMPLEMENTATION

SAMPLE CODE:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Present via Pixel — NTTF College</title>
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
    rel="stylesheet">
  <style>
    html, body {
      margin: 0;
      padding: 0;
      height: 100%;
      overflow: hidden;
      font-family: 'Poppins', sans-serif;
      background: radial-gradient(circle at 20% 20%, #007bff, #001f3f);
      color: white;
    }

    /* Navbar */
    .navbar {
      background: rgba(0, 0, 0, 0.9);
      position: fixed;
      top: 0;
      left: 0;
      right: 0;
      z-index: 1200;
      height: 65px;
      padding: 0 25px;
      display: flex;
      align-items: center;
```

```
justify-content: space-between;  
}
```

```
.navbar-brand {  
    display: flex;  
    align-items: center;  
    color: #fff !important;  
    text-decoration: none;  
    font-weight: 600;  
    font-size: 18px;  
}
```

```
.navbar-brand img {  
    height: 40px;  
    margin-right: 10px;  
}
```

```
.nav-actions {  
    display: flex;  
    align-items: center;  
    gap: 0.75rem;  
}
```

```
.nav-btn {  
    padding: 6px 14px;  
    border-radius: 999px;  
    font-size: 0.85rem;  
    text-decoration: none;  
    border: 1px solid transparent;  
    transition: 0.3s ease;  
    color: #e5edff;  
    cursor: pointer;  
    user-select: none;  
}
```

```
.nav-btn.nav-ghost {  
    border-color: rgba(255, 255, 255, 0.4);  
    background: transparent;  
}  
  
/* Hover for nav buttons */  
.nav-btn:hover {  
    background: rgba(255, 255, 255, 0.15);  
    color: #ffffff;  
    transform: translateY(-1px);  
    box-shadow: 0 0 10px rgba(0, 198, 255, 0.5);  
}  
  
/* Click (active) effect for nav buttons */  
.nav-btn:active {  
    transform: translateY(0);  
    box-shadow: 0 0 4px rgba(0, 0, 0, 0.6);  
    background: rgba(0, 180, 255, 0.3);  
}  
  
.nav-btn.nav-primary {  
    background: #00b4ff;  
    color: #001021;  
    font-weight: 600;  
    box-shadow: 0 8px 18px rgba(0, 0, 0, 0.35);  
}  
  
.nav-btn.nav-primary:hover {  
    background: #00d4ff;  
    color: #001021;  
    box-shadow: 0 0 18px rgba(0, 212, 255, 0.8);  
    transform: translateY(-2px);  
}
```

```
/* Hero Section */
.hero {
    height: calc(100vh - 70px);
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    text-align: center;
    padding: 0 20px;
}

.hero h1 {
    font-size: 3.2rem;
    font-weight: 700;
    background: linear-gradient(90deg, #ffffff, #00c6ff);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
    text-shadow: 0 0 20px rgba(0, 198, 255, 0.3);
    margin-bottom: 15px;
}

.hero p {
    font-size: 1.1rem;
    opacity: 0.9;
    margin-bottom: 28px;
}

.hero-small {
    font-size: 0.9rem;
    opacity: 0.8;
    margin-bottom: 35px;
}

/* Single main login button */
.hero-login-btn {
    width: 240px;
    padding: 14px;
```

```
font-size: 17px;
border-radius: 999px;
font-weight: 600;
background: #ffffff;
color: #007bff;
border: none;
transition: 0.3s ease;
box-shadow: 0 4px 15px rgba(0, 0, 0, 0.35);
text-decoration: none;
display: inline-block;
}

.hero-login-btn:hover {
background: linear-gradient(135deg, #00c6ff, #007bff);
color: #ffffff;
box-shadow: 0 0 25px rgba(0, 198, 255, 0.85);
transform: translateY(-3px);
}

/* Footer */
footer {
position: fixed;
bottom: 0;
width: 100%;
background: rgba(0, 0, 0, 0.9);
color: #ccc;
text-align: center;
padding: 10px 0;
font-size: 0.9rem;
}

@media (max-width: 768px) {
.navbar {
padding: 0 15px;
}
}
```

```

.hero h1 {
  font-size: 2.4rem;
}

.nav-actions {
  gap: 0.5rem;
}

.nav-btn {
  padding: 4px 10px;
  font-size: 0.75rem;
}

</style>
</head>
<body>

<!-- Navbar -->
<nav class="navbar">
  <a class="navbar-brand" href="#">
    
    NETTUR TECHNICAL TRAINING FOUNDATION
  </a>

  <div class="nav-actions">
    <a href="{{ url_for('static', filename='about.txt') }}" class="nav-btn nav-ghost">About</a>
    <a href="{{ url_for('static', filename='Features.txt') }}" class="nav-btn nav-ghost">Features</a>
    <a href="{{ url_for('static', filename='Help.txt') }}" class="nav-btn nav-ghost">Help</a>
    <a href="{{ url_for('login') }}" class="nav-btn nav-primary">Login</a>
  </div>
</nav>

```

```
<!-- Hero Section -->
<section class="hero">
  <h1>Present via Pixel</h1>
  <p>Where every presence is valued, and every moment counts.</p>
  <a href="{{ url_for('login') }}" class="hero-login-btn">
    Get Started
  </a>
</section>

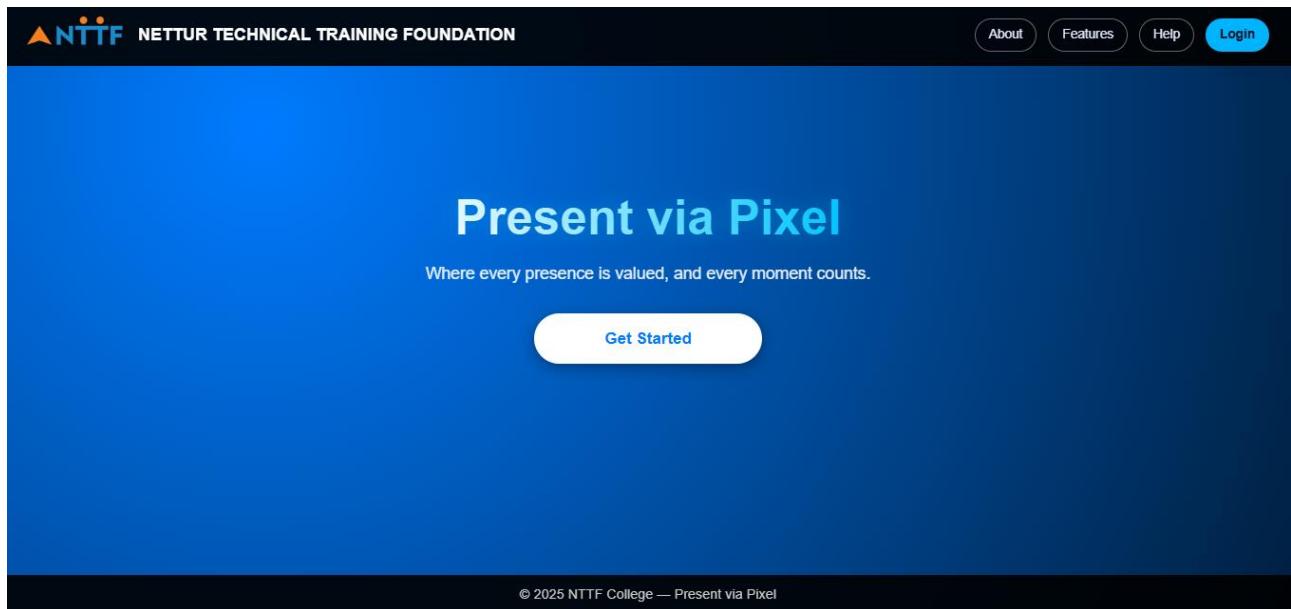
<footer>
  © 2025 NTTF College — Present via Pixel
</footer>

</body>
</html>
```

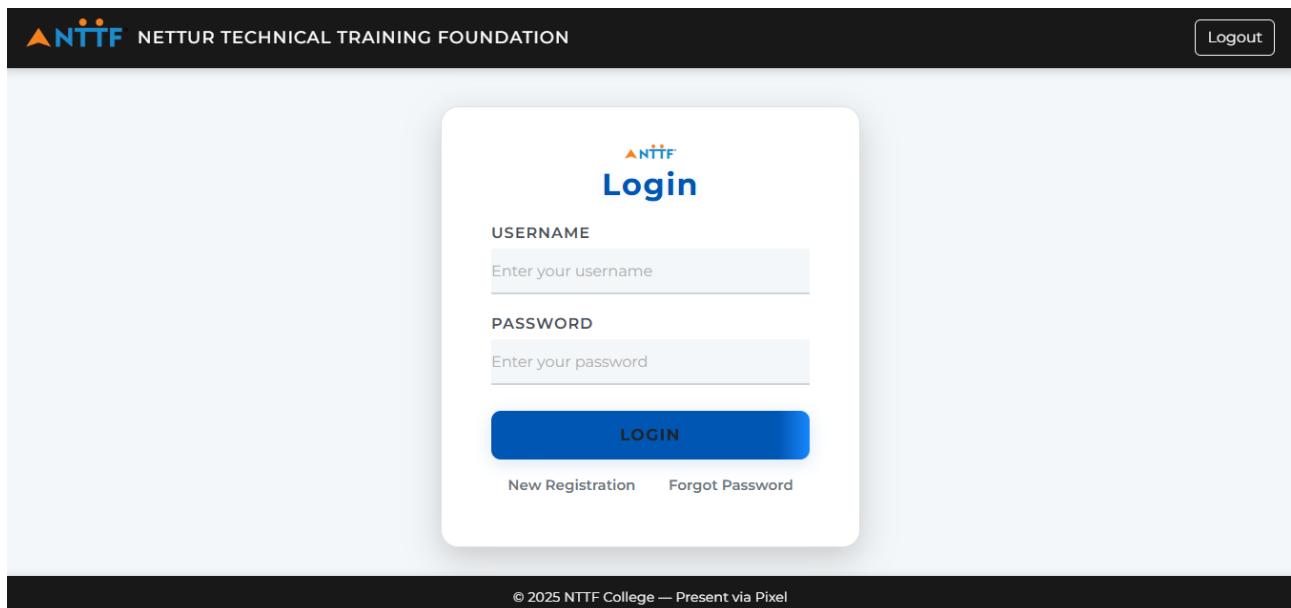
CHAPTER 7

RESULTS

HOME PAGE:



LOGIN PAGE:



STUDENT DASHBOARD:

NETTUR TECHNICAL TRAINING FOUNDATION

STUDENT

Overview of today's attendance and student performance metrics.

Total Students 7	Present Today 4	Absent Today 3	Average Attendance (%) 57.1%
----------------------------	---------------------------	--------------------------	--

Present Students - 2025-12-03

Token No	Name	Time In	Status
nec0923041	gunjan	08:28:47	Present
nec0923045	Aarti	08:28:55	Present
nec0923021	anchal	09:59:14	Present
nec0923039	Reena	10:00:10	Present

© 2025 NTTF College — Present via Pixel

ADMIN BASHBOARD:

NETTUR TECHNICAL TRAINING FOUNDATION

ADMIN

Overview of today's attendance and student performance metrics.

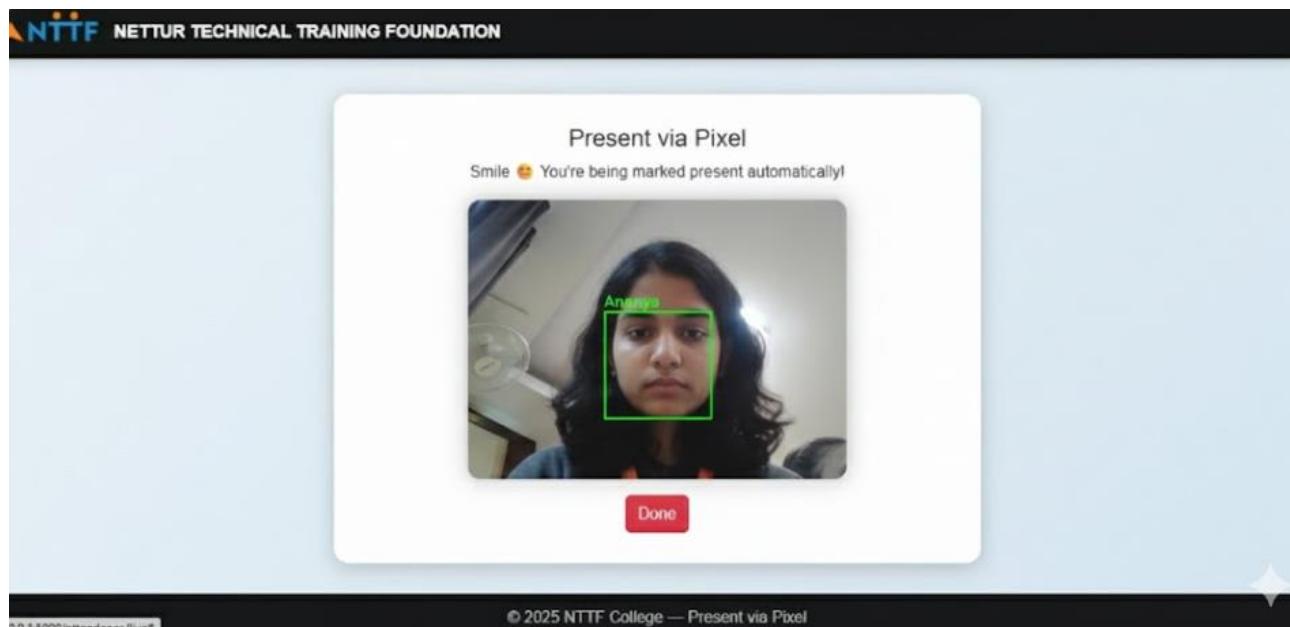
Total Students 7	Present Today 4	Absent Today 3	Average Attendance (%) 57.1%
----------------------------	---------------------------	--------------------------	--

Present Students - 2025-12-03

Token No	Name	Time In	Status
nec0923041	gunjan	08:28:47	Present
nec0923045	Aarti	08:28:55	Present
nec0923021	anchal	09:59:14	Present
nec0923039	Reena	10:00:10	Present

© 2025 NTTF College — Present via Pixel

MARK ATTENDANCE:



MANAGE STUDENTS:

TOKEN NO	NAME	ACTIONS
nec0923045	Aarti	<button>EDIT</button> <button>DELETE</button>
nec0923039	Reena	<button>EDIT</button> <button>DELETE</button>
nec0923021	anchal	<button>EDIT</button> <button>DELETE</button>
nec0923041	gunjan	<button>EDIT</button> <button>DELETE</button>
nec0923046	manisha	<button>EDIT</button> <button>DELETE</button>
nec0923035	sheetal	<button>EDIT</button> <button>DELETE</button>
nec0923037	Vanshika	<button>EDIT</button> <button>DELETE</button>

© 2025 NTTF College — Present via Pixel

REGISTER NEW STUDENTS:

←  NETTUR TECHNICAL TRAINING FOUNDATION

Register New Student

Full Name:

Token No (Unique ID):

Face Capture:

[START](#)

[CAPTURE](#)

[REGISTER](#)

© 2025 NTTF College — Present via Pixel

VIEW ATTENDANCE:

←  NETTUR TECHNICAL TRAINING FOUNDATION

View Attendance

03-12-2025

[Load](#)
[CSV](#)
[Excel](#)

Token No	Name	Date	Time	Status
nec0923041	gunjan	2025-12-03	08:28:47	On Time
nec0923045	Aarti	2025-12-03	08:28:55	On Time
nec0923021	anchal	2025-12-03	09:59:14	Late
nec0923039	Reena	2025-12-03	10:00:10	Late

Absent Students

Token No	Name	Date	Status
nec0923046	manisha	2025-12-03	Absent
nec0923035	sheetal	2025-12-03	Absent
nec0923037	varsha	2025-12-03	Absent

© 2025 NTTF College — Present via Pixel

CHAPTER 8

CONCLUSION

Present via Pixel delivers a fast, secure and fully automated attendance solution using face recognition and liveness detection instead of manual roll calls. It reduces proxy attendance, eliminates paper registers, and gives NTTF a reliable, real-time record of every student's presence in the campus.

Key features:

- Face-based, contactless attendance using live camera and geofence-verified location.
- Liveness detection (blink and head-movement checks) to stop photo or video spoofing.
- Smart duplicate blocking ensures only one valid attendance per student per day.
- Role-based dashboards for admin, teacher and student with secure login and access.
- Attendance analytics and export to CSV/Excel for reports, audits and performance review.

REFERENCES:

- www.youtube.com
- <https://www.geeksforgeeks.org>
- <https://github.com>