Abstract:

This project utilizes object and face recognition technology to automate attendance tracking and authentication. The system captures images from a CCTV camera installed in a classroom or office to recognize individuals and generate attendance reports. It enhances efficiency in educational institutions and workplaces by eliminating manual attendance marking.

Key Features & Responsibilities:

- The system captures images or video feeds from a CCTV camera installed in a classroom, office, or hospital entrance.
- The system uses OpenCV and deep learning models to detect faces and classify individuals based on pre-trained datasets.
- Object recognition algorithms distinguish between different categories (e.g., humans, employees, patients).
- The detected faces are compared with stored profiles in the database using face embedding and feature extraction techniques.
- If a match is found, the system identifies the person and proceeds with the respective functionality (attendance marking, clock-in/out, or patient authentication).
- Once a student or employee is recognized, the system logs their attendance into the database without requiring manual input.
- The attendance report is automatically generated and stored for later reference.
- Employees are identified at entry and exit points to log their working hours automatically.
- The system stores clock-in/out timestamps, reducing manual tracking errors. When a patient arrives, the system identifies their face and pulls up their medical records.
- The system provides instant access to patient history, prescriptions, and appointments, reducing waiting time.
- Attendance records, employee work hours, and patient details are securely stored in MySQL
- Authentication is encrypted to prevent unauthorized access to sensitive records.
- The system can be linked to HRMS and HMS software to streamline operations and enhance workplace efficiency.
- Reports are generated automatically for attendance tracking and patient visits.
- Notifications can be sent to administrators for missing attendance or unauthorized access.

Technologies Used:

- Programming Language: Python, MATLAB
- Libraries & Frameworks: OpenCV, TensorFlow, Keras, dlib (for face recognition)
- Database: MySQL, MongoDB (for storing records)
- Hardware: CCTV Cameras, Image Processing Units
- Techniques: Facial Recognition, Object Detection, Machine Learning, Deep Learning

Abstract:

The Flight Ticketing System is a Java-based web application that allows users to book flight tickets while providing real-time weather updates for the selected destination. Built using Java, Servlets, and MySQL, the system enables users to search for available flights, select seats, and complete bookings. The integration of weather updates enhances the user's experience by providing relevant travel conditions. The project follows Object-Oriented Programming (OOP) principles to ensure modularity and scalability.

Key Features & Responsibilities:

- Designed and developed a flight ticketing system using Java and Servlets. Implemented real-time weather updates using an external API to provide weather forecasts for flight destinations.
- Developed dynamic web pages using HTML and CSS for interactive user experience.
- Generates a booking confirmation with ticket details, including flight information and seat assignment. A simulated payment gateway for completing bookings. Users can view their past and upcoming bookings.
- Utilized MySQL to store flight details, user bookings, and payment information.
- Applied OOP concepts such as inheritance, polymorphism, and encapsulation to improve code structure.
- Admins can add new flights, update schedules, and manage bookings. Users can filter flights based on price, timing, and airlines. The web application is designed using HTML & CSS, ensuring compatibility across different devices.
- Managed session handling for users to maintain login state and booking progress.
- Implemented form validation and error handling to ensure data integrity. Optimized SQL queries for efficient database interactions in MySQL.
- Users can create accounts, log in, and manage their bookings. Users can search for flights based on destination, date, and availability, and book tickets. Integrated a weather API to display the latest weather conditions for the selected flight destination. Allows users to choose their preferred seats before confirming the booking.

Technologies Used:

- Backend: Java, Servlets, OOP Concepts
- Database: MySQL
- Frontend: HTML, CSS
- Other: Weather API (for live updates), Session Management

Abstract:

This project involves the design and simulation of multi-layered chips using CMOS and NMOS logic gates to perform multi-level calculations efficiently. The chips were designed using Logisim, Multisim, and Mentor Graphics tools, ensuring optimal power consumption and high-speed computation. The project focuses on circuit optimization, transistor-level logic implementation, and verification of logical operations using simulation tools.

Key Features:

- Developed multiple layers of logic gates using CMOS and NMOS technology. Optimized circuits to reduce power consumption while maintaining high-speed processing. Designed circuits capable of performing arithmetic, logic, and multi-level calculations.
- Implemented MOSFET-based logic gates and verified their switching characteristics. Used Logisim, Multisim, and Mentor Graphics to simulate the circuit behavior and validate performance. Designed chips with a focus on scalability for complex calculations and multi-step processing.
- Implemented logic gates using CMOS and NMOS transistors for better speed and power efficiency.
- Designed multi-layered architecture where each layer performs a different level of calculations (e.g., arithmetic operations, logic operations).
- Simulated circuit performance using Logisim, Multisim, and Mentor Graphics to ensure functional correctness. Verified voltage levels, switching times, and power dissipation to optimize chip performance.
- Designed chips to execute sequential and parallel calculations for improved efficiency. Ensured low power consumption and high-speed operations using optimized logic designs.
- Conducted truth table validation for all logic circuits. Measured delay times, propagation speed, and transistor switching behavior.
- Suggested applications in embedded systems, signal processing, and high-speed computing.

Technologies & Tools Used:

- Circuit Design & Simulation Tools: Logisim, Multisim, Mentor Graphics
- Logic Families Used: CMOS (Complementary Metal-Oxide-Semiconductor) and NMOS (N-type Metal-Oxide-Semiconductor)
- Digital Logic Components: AND, OR, XOR, NAND, NOR, Inverters
- Arithmetic Circuits: Adders, Subtractors, Multipliers
- Multi-Level Calculation Logic: Sequential and combinational logic circuits

Abstract:

This project involves the design and implementation of a gesture-controlled smart glove using accelerometer sensors to interact with home electronic devices. The glove detects specific hand gestures and translates them into commands to control appliances such as lights, fans, and other smart devices. The system uses embedded programming, microcontrollers, and wireless communication to provide an intuitive and efficient way to control home electronics.

Key Features:

- Detects predefined hand gestures using an accelerometer sensor. Sends control signals wirelessly via Bluetooth or RF module to home automation devices.
- Users can configure different gestures for specific electronic appliances. Utilized Arduino or Raspberry Pi to process sensor data and execute commands. Can be integrated with smart home systems and IoT devices.
- Designed for efficient power management for extended battery life. The system processes gestures instantly to provide quick responses.
- The glove is lightweight and comfortable, making it easy to use for daily tasks. The system can be extended to control multiple appliances and integrated with voice assistants.
- Can be adapted for disabled individuals to control their environment more easily.
- The accelerometer sensor detects hand movements in X, Y, and Z axes. Predefined gestures are mapped to specific electronic device commands.
- The sensor data is processed using Arduino / Raspberry Pi. Based on gesture recognition, a control signal is generated. The control signal is transmitted wirelessly via Bluetooth or RF module.
- The receiver connected to home appliances interprets the signal and executes the action. The relay module toggles the power state of electronic devices based on received commands.
- Ensures low latency for quick response time. Can be modified for gesture customization using a machine learning-based gesture recognition model.

Technologies & Components Used:

- Microcontroller: Arduino / Raspberry Pi
- Sensors: 3-Axis Accelerometer ADXL345
- Wireless Communication: Bluetooth Module (HC-05) / RF Module
- Power Source: Rechargeable Battery / Power Bank
- Programming Languages: Embedded C, Python
- Control System: Relays for switching electrical appliances
- IoT Integration: Can be connected to smart home systems via MQTT or WiFi module (ESP8266)