



SRUA

COMPLETE PROJECT MARKETPLACE

Ready-made Engineering, Diploma & Computer Science
Projects with Source Code, Reports, Presentations &
Documentation.

ACADEMIC PROJECT CATALOG • 2026 EDITION

Verified Academic Projects

SRUA is an elite, dedicated project marketplace designed explicitly for engineering, diploma, and computer science students. We recognize the challenges of balancing coursework, learning complex architectures, and preparing submissions. To address this, we compile ready-made project repositories across high-demand technical domains.

Every single project listed in our catalog is fully optimized and thoroughly verified for local environment execution. We include complete, clean, modular source code, structured documentation, professional project reports, ready-to-present PPT slides, and system setup scripts.

SUPPORTED CORE TECHNOLOGIES

- AI & ML
- MERN Stack
- Python
- Java & C++
- React
- Node.js
- Android
- MySQL
- Blockchain
- IoT
- Cloud
- Data Science
- OpenCV
- OpenCV OCR
- TensorFlow
- FastAPI

Academic-Ready Benefits

Why thousands of students trust SRUA for project resource deliverables



Verified Code

Every repository contains fully executable source code verified locally.



Easy Setup

Step-by-step setup guides to launch projects on your local machine instantly.



Complete Docs

Comes with full reports, PPT presentations, and system architecture plans.



Latest Tech

Projects built with modern libraries like TensorFlow, FastAPI, FAISS, and React.



Academic Ready

Meets all standard evaluation metrics for engineering & diploma panels.



Easy to Learn

Clean, well-commented modules designed to help students study easily.



Database Scripts

Includes complete DB mocks (MongoDB, MySQL, SQLite) for seamless setups.



Regular Updates

Codebases are actively maintained for current compiler/library versions.



Support Included

Receive deployment and setup guidance to resolve launch roadblocks.

Core Technology Streams

Explore academic projects tailored across mainstream fields (Part 1)



AI & Machine Learning

Predictive models, neural networks, and automated reasoning pipelines.



Cybersecurity

Data protection, secure protocols, network scanners, and cryptography.



Python

Data processing, algorithms, system scripts, and ML automation.



Java

Object-oriented desktop apps, enterprise patterns, and core systems.



C++

High-performance processing, game elements, and hardware interaction.



MERN Stack

Full-stack apps utilizing React frontend and Node/Express/Mongo backend.



React

Modern UI rendering, state management, and responsive interfaces.



Node.js

Scalable backend routing, API services, and microservice structures.



Android

Native mobile tools, local storage databases, and responsive views.



Web Development

Responsive, accessible web structures using clean code guidelines.

Specialized Categories

Explore academic projects tailored across mainstream fields (Part 2)



Blockchain

Smart contracts, decentralized ledgers, and token transactions.



IoT

Hardware sensor reading, serial bridges, and visual metric logs.



Cloud Computing

Distributed storage setups, backend API deployment configurations.



Data Science

Exploratory statistics, CSV cleaners, and interactive dashboard graphs.



Computer Vision

Image manipulation, object tracking, and OpenCV video capture feeds.



NLP

Text preprocessing, skill tokenizers, and similarity prediction models.



Final Year Projects

Complex systems matching advanced university coursework requirements.



Mini Projects

Focused project repositories suited for intermediate assessments.



Major Projects

End-to-end full stack software systems with deep integrations.



Diploma Projects

Hands-on configurations focusing on practical, core technology.



Engineering Projects

Advanced engineering projects built on verified codebases.

What Every Project Includes

A comprehensive checklist of files packaged inside each SRUA repository bundle



Complete Source Code

Fully tested, structured, and ready-to-run repository folder.



Project Report

Pre-written doc/pdf files containing objectives, methodology, and results.



Documentation

Detailed technical specifications, environment parameters, and dependencies.



PPT Slides

Ready presentation templates outlining features, modules, and workflows.



Database Scripts

Complete SQL / MongoDB files to mock application storage instantly.



Setup Guide

Step-by-step local host execution commands and version lists.



Screenshots

Visual verification prints of user interfaces and API endpoints.



Architecture Diagram

Clear structural flowcharts showing server, DB, and client modules.



Flowchart

Logic maps tracking execution controls and query conditions.



ER Diagram

Entity relationships modeling data tables and database fields.



Implementation Guide

Best-practice deployment setups and compiler tips.



README File

Markdown instruction layout containing setup guidelines.

#01

AI & MACHINE LEARNING

RAGSmith

FastAPI

FAISS

Ollama

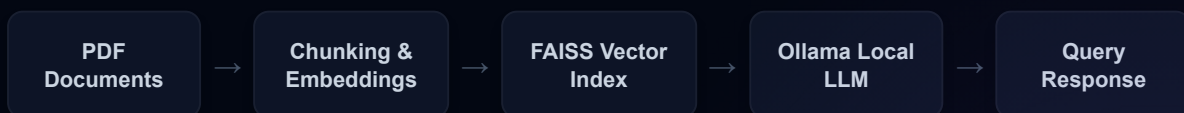
Sentence Transformers

Python

RAGSmith is a fully offline Retrieval-Augmented Generation (RAG) platform engineered for building local AI knowledge assistants. It leverages FastAPI as the high-performance API gateway, FAISS as the in-memory vector similarity search engine, Sentence Transformers for generating dense semantic embeddings from documents, and Ollama to run quantised large language models entirely on your own hardware — no cloud API key required. Users can upload PDF or text documents, which are automatically chunked, embedded, and indexed into a FAISS vector store. At query time, the system retrieves the most semantically relevant document chunks and injects them as context into the local LLM, ensuring precise, grounded responses rather than hallucinated outputs. Ideal for enterprises, researchers, and students who need AI-powered document search with full data privacy.

 **Difficulty: Advanced** **Suitable for: B.E. / B.Tech / MCA**

SYSTEM DATA FLOW / ARCHITECTURE



#02

COMPUTER VISION

HeadMouse

MediaPipe

OpenCV

Python

PyAutoGUI

HeadMouse is an innovative accessibility tool that transforms facial movements and hand gestures into precise cursor control actions — eliminating the need for a physical mouse entirely. The system captures a real-time webcam stream and feeds each frame into MediaPipe FaceMesh, which detects 468 facial landmark coordinates with sub-pixel accuracy. The nose tip coordinates are mapped to a calibrated screen space using OpenCV affine transformations, while PyAutoGUI translates those coordinates into smooth operating system cursor events. Eye blink detection powered by the Eye Aspect Ratio (EAR) algorithm is used as the click trigger. This project is an outstanding demonstration of applied computer vision for human-computer interaction (HCI) and serves as an excellent final-year project for students interested in assistive technology, accessibility engineering, and real-time pose estimation.

 **Difficulty: Advanced** **Suitable for: B.E. / B.Tech / BCA**

SYSTEM DATA FLOW / ARCHITECTURE

Camera Stream

MediaPipe
FaceMeshEye/Nose
CoordinatesCursor
TranslationBlink/Click
Event

#03

MERN STACK

MediMage

React

Node.js

Express

MongoDB

Tailwind CSS

MediMage is a full-stack Doctor Appointment Management System built on the industry-standard MERN architecture. The platform provides role-based authentication for both patients and doctors, complete with JWT-secured sessions and bcrypt password hashing. Patients can browse available doctors by specialisation, view real-time availability calendars, and book or cancel appointments through an intuitive dashboard. Doctors receive instant notifications of new bookings and can manage their schedule, update consultation notes, and view patient histories. The Express backend exposes a clean RESTful API consumed by the React frontend, while MongoDB stores all appointment and user data in flexible document schemas. A responsive UI built with Tailwind CSS ensures accessibility across desktop and mobile devices. This project demonstrates expertise in modern full-stack development, RESTful API design, and database modelling.

 **Difficulty: Medium** **Suitable for: Diploma / BCA / B.Tech**

SYSTEM DATA FLOW / ARCHITECTURE

User Login

Dashboard
BookingExpress API
RouterMongoDB
DatabaseEmail
Confirmation

#04

WEB DEVELOPMENT

AlumniConnect

React

Node.js

Express

MongoDB

Socket.io

AlumniConnect is a comprehensive digital platform that bridges the gap between current students and alumni of educational institutions, fostering a culture of mentorship, peer learning, and career development. Students can create detailed profiles highlighting their skills and career aspirations, while alumni can register with their professional experience, industry, and willingness to mentor. The platform features an intelligent mentor-matching algorithm that pairs students with the most relevant alumni based on domain, skill overlap, and availability. Real-time communication is powered by Socket.io, enabling instant messaging between matched pairs without page refreshes. The system also includes a public job and internship board where alumni can post verified opportunities exclusively available to students from their institution. A notification engine keeps users updated on new messages, mentorship requests, and job postings. This project is a strong showcase of real-time web application architecture and social networking backend design.

 **Difficulty: Medium** **Suitable for: B.Tech / MCA / BCA**

SYSTEM DATA FLOW / ARCHITECTURE

Student Portal

Alumni
DirectoryChat / Mail
RequestMentorship
Matching

Socket.io Chat

#05

WEB DEVELOPMENT

FoodBridge

React

Node.js

Express

MongoDB

Leaflet Maps

FoodBridge is a social impact platform engineered to combat food wastage by creating a real-time bridge between surplus food donors — including restaurants, caterers, and households — and verified NGOs and food distribution agencies. Donors post available food listings with details including item type, quantity, preparation time, and a geographic pickup location visualised on an interactive Leaflet map. The backend implements a geospatial proximity query using MongoDB's 2dsphere indexing, which automatically notifies nearby registered NGOs within a configurable radius. Once an NGO claims a food post, the system locks it and logs the delivery chain to ensure accountability. The dashboard provides statistics on total food diverted from waste, number of meals facilitated, and active donors in the network. The project demonstrates geospatial querying, real-time notification systems, and social-good application development — powerful themes for final-year project evaluation panels.

 **Difficulty: Medium** **Suitable for: B.Tech / Diploma**

SYSTEM DATA FLOW / ARCHITECTURE

Donor Posts
FoodGeographic
Radius MapNGO
NotificationClaim Food
PostDelivery
Logged

#06

BACKEND SYSTEMS

Policy Engine

FastAPI

MongoDB

Python

Docker

The Policy Engine is a production-grade dynamic rule evaluation system designed for enforcing access control, compliance checks, and automated business logic decisions at runtime. Administrators define policy rules as structured JSON documents stored in MongoDB — rules can be composed using logical operators (AND, OR, NOT), condition matchers, and attribute comparisons without requiring code redeployment. The FastAPI backend exposes an evaluation endpoint that accepts an input context object, retrieves relevant rules from MongoDB, parses them using an internal Abstract Syntax Tree (AST) interpreter, evaluates the conditions recursively, and returns a deterministic ALLOW or DENY decision within milliseconds. The entire system is containerised with Docker for reproducible deployment across any environment. This project is particularly relevant for students interested in backend systems engineering, policy-as-code architectures, and enterprise access management platforms.

 **Difficulty: Advanced** **Suitable for: B.Tech / B.E. / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

JSON Rules
Input

FastAPI Parse

MongoDB
Rules LookupAST Rule
EvaluationAllow/Deny
Result

#07

WEB DEVELOPMENT

Pet Care

React

Node.js

Express

MySQL

Pet Care is a fully functional, end-to-end pet service management platform designed for veterinary clinics, grooming centres, and pet boarding facilities. Pet owners can register their animals with health records, vaccination histories, and dietary preferences, then book a range of services including medical consultations, grooming sessions, training packages, and boarding stays. The Express backend manages all service scheduling through a relational MySQL database, ensuring conflict-free appointment slots and automated billing generation. Service providers get access to a staff dashboard displaying upcoming bookings, pet medical histories, and daily revenue summaries. The React frontend delivers a clean, intuitive interface with real-time availability indicators. This project showcases relational database design with normalised tables, RESTful service architecture, and multi-role user authentication — all essential skills demonstrated effectively for academic and professional evaluation.

 **Difficulty: Easy** **Suitable for: Diploma / BCA**

SYSTEM DATA FLOW / ARCHITECTURE

Pet Owner
ProfileService
Schedule

Express API



MySQL Storage



Billing Receipt

#08

FRONTEND UI

YouTube Clone

HTML5

CSS3

JavaScript

The YouTube Clone is a pixel-accurate, fully responsive frontend replica of the YouTube platform built entirely using vanilla HTML5, CSS3, and JavaScript — demonstrating the power of core web technologies without any framework dependencies. The layout is constructed using a combination of CSS Grid and Flexbox, faithfully reproducing YouTube's homepage structure including the top navigation bar with a search field, a collapsible sidebar with category links, and a responsive video card grid that adapts gracefully from mobile to widescreen desktop resolutions. Each video card displays a thumbnail, title, channel name, view count, and upload time in a format identical to the real platform. Embedded video playback is supported using the HTML5 video element. This project is the perfect starting point for students learning frontend fundamentals, demonstrating a thorough understanding of document structure, CSS layout systems, media queries, and the DOM API.

 **Difficulty: Easy** **Suitable for: Diploma / First Year**

SYSTEM DATA FLOW / ARCHITECTURE

Static HTML
IndexCSS Grid
Layout

Flex Navigation

Video Embed
PlayerSidebar
Collapse

#09

NATURAL LANGUAGE PROCESSING

AI Chatbot

Python

NLTK

TensorFlow

Flask

The AI Chatbot is an intelligent conversational agent built using Natural Language Processing (NLP) techniques and a custom-trained deep neural network. User messages are first processed through an NLTK tokenization and stemming pipeline, reducing words to their root forms and removing stop words. Each processed message is converted into a Bag-of-Words (BoW) feature vector, which is fed into a TensorFlow-powered Deep Neural Network (DNN) classifier trained on a curated dataset of labelled intent patterns. The model predicts the most likely user intent from a predefined set — such as greetings, FAQs, product queries, or farewells — and retrieves a contextually appropriate response from a structured JSON response library. The Flask web server exposes the chatbot through a clean, real-time chat interface. This project is a comprehensive demonstration of NLP pipeline construction, intent classification, neural network training, and API deployment — all highly relevant topics in modern AI applications.

 **Difficulty: Medium** **Suitable for: B.Tech / BCA / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

User Message

NLTK Tokenize
& StemBag-of-Words
VectorTensorFlow
DNN ModelJSON
Response

#10

MACHINE LEARNING

Disease Prediction

Python

Scikit-Learn

Flask

Pandas

The Disease Prediction System is a machine learning-powered health assessment application that helps users identify potential medical conditions based on self-reported symptoms. Users select their symptoms from a comprehensive list, and the application encodes these inputs as one-hot feature vectors. A Random Forest classifier trained on a curated medical symptom-disease dataset then predicts the most probable diagnosis with a confidence score. The system further enhances its utility by cross-referencing the predicted disease with a specialist recommendation database, guiding users toward the appropriate type of medical professional. The Flask web interface provides a clean, step-by-step symptom selection experience with instant prediction results. Pandas is used extensively for data preprocessing, feature engineering, and dataset cleaning. This project demonstrates the complete supervised machine learning pipeline — from raw data to a deployable health AI application — making it an outstanding submission for biomedical informatics or data science specialisations.

 **Difficulty: Medium** **Suitable for: B.Tech / B.E. / MCA**

SYSTEM DATA FLOW / ARCHITECTURE



#11

MACHINE LEARNING

Movie Recommendation System

Python

Pandas

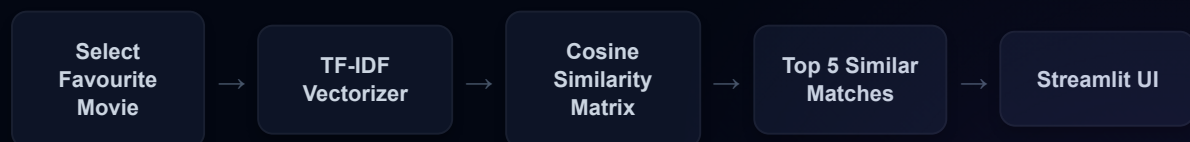
Scikit-Learn

Streamlit

The Movie Recommendation System is a content-based filtering engine that analyses movie metadata to surface personalised viewing suggestions. The system ingests a large movie dataset containing titles, genres, cast, director, and keywords. These attributes are combined into a comprehensive content profile for each film, and a TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer transforms the profiles into high-dimensional numerical vectors. Cosine similarity scores are then computed across all movie vectors to measure content overlap, creating a full similarity matrix. When a user selects a favourite movie, the system retrieves the top five most similar films ranked by their cosine similarity scores and presents them with poster thumbnails and metadata through an elegant Streamlit web interface. This project illustrates core recommendation algorithm design, natural language feature extraction, and the construction of interactive data-driven web applications — skills that are highly sought after in data science and AI engineering roles.

 **Difficulty: Medium** **Suitable for: B.Tech / BCA / MCA**

SYSTEM DATA FLOW / ARCHITECTURE



#12 Resume Shortlisting System

MACHINE LEARNING

Python

NLTK

Scikit-Learn

Streamlit

The Resume Shortlisting System is a machine learning-powered Applicant Tracking System (ATS) that automates the screening and ranking of candidate resumes against a defined job description. HR professionals upload a batch of PDF resumes and provide a target job description. The system uses the NLTK library to extract and clean text from each resume, stripping formatting artifacts and stop words. A TF-IDF Vectorizer then converts both resumes and the job description into high-dimensional feature vectors, capturing the relative importance of keywords across the corpus. A K-Nearest Neighbours (KNN) classifier computes similarity scores between the job vector and each resume vector, producing a ranked shortlist of candidates. Results are presented through a Streamlit dashboard with downloadable CSV exports. This project is particularly valuable for students interested in HR tech, NLP automation, and building enterprise-grade document processing pipelines.

 **Difficulty: Advanced** **Suitable for: B.Tech / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

Upload PDF
ResumesNLTK Text
ExtractionTF-IDF Skill
VectorsK-Nearest
Neighbours
ModelRanked Score
List

#13 **Driver Drowsiness Detection**

COMPUTER VISION

OpenCV

Dlib

Python

Pygame

The Driver Drowsiness Detection system is a real-time road safety application that continuously monitors a driver's alertness level using facial landmark analysis and triggers an audio alarm before a fatigue-related accident can occur. The system processes a live video feed from a dashboard-mounted webcam using OpenCV. The Dlib facial landmark predictor identifies 68 precise landmark points on the driver's face in each frame. The Eye Aspect Ratio (EAR) — a mathematical formula derived from the vertical and horizontal distances between eyelid landmarks — is computed for both eyes. When the EAR value drops below a pre-calibrated threshold for a sustained number of consecutive frames, indicating that the eyes are closing due to drowsiness, the system immediately triggers a loud audio alarm through the Pygame mixer to wake the driver. This system is applicable to automotive safety systems, fleet management solutions, and IoT-connected vehicle monitoring platforms.

 **Difficulty: Advanced** **Suitable for: B.E. / B.Tech / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

Video Capture
FeedDlib Facial
LandmarksEye Aspect
Ratio (EAR)
CalcThreshold
CheckPygame Alarm
Ring

#14

COMPUTER VISION

Automatic Number Plate Recognition

OpenCV

Tesseract OCR

Python

The Automatic Number Plate Recognition (ANPR) system is a computer vision pipeline designed to detect, localise, and extract alphanumeric text from vehicle license plates in real-time video streams or static images. The processing pipeline begins by converting the input frame to grayscale and applying a bilateral filter to preserve edge sharpness while removing noise. Canny edge detection identifies contour candidates, from which rectangular regions matching the aspect ratio of a standard license plate are isolated. The cropped plate region is then passed through Tesseract OCR's neural network-powered character recognition engine with custom plate-optimised settings, extracting the license number as a structured string. The recognised plate numbers are logged with timestamps to a database for vehicle access control or traffic monitoring use cases. This project demonstrates the full computer vision engineering pipeline — from image acquisition and preprocessing to OCR-based text extraction — and is directly applicable to smart city and traffic management systems.

 **Difficulty: Advanced** **Suitable for: B.E. / B.Tech / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

Vehicle Image
InputGray Scaling &
Bilateral FilterCanny Edge
Plate CropTesseract OCR
ParseLicense
Number String

#15

COMPUTER VISION

Face Mask Detection

TensorFlow

Keras

OpenCV

MobileNetV2

The Face Mask Detection system is a real-time deep learning application that automatically identifies whether individuals in a live video stream are wearing protective face masks — a capability with direct applications in public health enforcement, workplace safety monitoring, and access control systems. The pipeline begins with a Caffe-based deep learning face detector that identifies and localises all human faces within each video frame, even under varying lighting conditions and partial occlusions. Each detected face region is then cropped and fed into a MobileNetV2-based binary classifier fine-tuned on a labelled mask/no-mask dataset. The lightweight MobileNetV2 architecture is specifically chosen for its high accuracy-to-performance ratio, enabling real-time classification at 30+ frames per second on standard hardware. Detection results are overlaid on the video feed as coloured bounding boxes — green for mask-compliant individuals and red for violations — along with confidence scores. This project exemplifies transfer learning, real-time object detection, and practical deployment of deep learning models.

 **Difficulty: Medium** **Suitable for: B.Tech / BCA / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

Live Webcam
StreamCaffe Face
Detector BoxMobileNetV2
Mask ClassifierMask / No Mask
AlertRGB Bound
Border

#16

DEEP LEARNING

Plant Disease Detection

TensorFlow

CNN

Python

Streamlit

The Plant Disease Detection system is a deep learning-powered agricultural intelligence tool that enables farmers, agronomists, and researchers to instantly identify crop diseases from leaf photographs. A Convolutional Neural Network (CNN) trained on the PlantVillage dataset — containing over 54,000 labelled images across 38 plant disease categories — learns to recognise subtle visual patterns in leaf texture, colour, and morphology that indicate specific pathogenic infections. Users upload a photograph of a diseased plant leaf through the Streamlit web interface, and the system preprocesses the image by resizing, normalising, and feeding it through the CNN. The final softmax output layer returns probability scores across all disease classes, and the system displays the predicted disease name along with its confidence percentage and recommended remediation measures. This project is a powerful example of applied deep learning for precision agriculture and is particularly relevant for students interested in AI for social impact and food security.

 **Difficulty: Advanced** **Suitable for: B.Tech / B.E. / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

Upload Leaf
PhotoTensorFlow
CNN LoadFeature Map
ExtractionSoftmax
Probability
ClassDisease Label
& Remedy

#17

HUMAN-COMPUTER INTERACTION

Hand Gesture Recognition

MediaPipe

OpenCV

Python

TensorFlow

The Hand Gesture Recognition system is a cutting-edge human-computer interaction (HCI) application that enables touchless control of digital interfaces through real-time recognition of hand gestures captured by a standard webcam. MediaPipe Hands — Google's state-of-the-art hand landmark detection model — tracks 21 precise three-dimensional keypoints across the palm and fingers in every frame of the video stream. These keypoint coordinates are normalised and assembled into a 63-dimensional feature vector representing the geometric configuration of the hand. A TensorFlow classifier trained on a custom gesture dataset maps these feature vectors to labelled gesture classes such as thumbs up, peace sign, fist, open palm, and pinch. Recognised gestures are mapped to configurable application actions — such as slide navigation, volume control, or media playback — through PyAutoGUI event triggers. This project demonstrates the integration of computer vision, pose estimation, custom dataset collection, and neural network classification into a cohesive real-time system.

 **Difficulty: Advanced** **Suitable for: B.E. / B.Tech / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

Webcam
Capture FrameMediaPipe
Hands
Extraction21 Keypoint
CoordinatesGesture Profile
MatchingAssigned
Action Trigger

#18

DATA SCIENCE

Restaurant Popularity Prediction

Python

XGBoost

Pandas

Flask

The Restaurant Popularity Prediction system is a data science application that uses gradient boosting machine learning to forecast the commercial popularity and customer demand potential of a restaurant based on its profile attributes. The system processes a rich feature set including geographic coordinates (decoded using Google Geolocation API), cuisine type, average meal cost, number of seating covers, restaurant category, and online order availability. Pandas handles extensive data cleaning, feature engineering, and one-hot encoding of categorical variables. The XGBoost regression model — an ensemble of decision trees trained via gradient boosting — achieves high predictive accuracy in estimating a composite popularity score. The Flask web interface provides a simple form for inputting restaurant parameters and returns a predicted popularity rating along with a comparative percentile ranking against similar establishments in the dataset. This project is ideal for students demonstrating skills in data preprocessing, feature engineering, ensemble modelling, and business analytics application development.

 **Difficulty: Medium** **Suitable for: B.Tech / BCA / MCA**

SYSTEM DATA FLOW / ARCHITECTURE

Restaurant
Profile InputGeolocation
CoordinatesCuisine & Cost
EncodingXGBoost
Regression
ModelPopularity
Rating Output

#19

COMPUTER VISION

Virtual Mouse

OpenCV

MediaPipe

Python

PyAutoGUI

The Virtual Mouse is a touchless input control system that replaces the physical computer mouse with hand gestures tracked through a standard webcam — offering a compelling demonstration of gesture-based human-computer interaction. The system captures a continuous video stream and feeds each frame into the MediaPipe Hands model, which tracks the three-dimensional positions of 21 hand landmarks in real time. The index finger tip landmark coordinates are extracted and mapped from the camera frame resolution to the full screen resolution using a linear scaling transformation, enabling smooth and precise cursor movement. A pinch gesture — detected by measuring the Euclidean distance between the index finger tip and thumb tip landmarks — triggers a simulated mouse click through PyAutoGUI. Additional gestures enable right-click, double-click, and scroll functionality. The system runs entirely on CPU without any specialised hardware, making it accessible and reproducible. This project demonstrates the seamless fusion of computer vision and operating system API interaction.

 **Difficulty: Advanced** **Suitable for: B.Tech / BCA**

SYSTEM DATA FLOW / ARCHITECTURE

Camera Stream
CaptureMediaPipe
Hand TrackingIndex Finger
Tip XYZScreen
Mapping
TranslationPinch Gesture
Click

#20

COMPUTER VISION

Face Recognition Attendance System

OpenCV

Face-Recognition

Python

SQLite

The Face Recognition Attendance System is a fully automated, contactless biometric attendance solution designed to eliminate manual roll-call processes in educational institutions and corporate environments. During enrollment, registered users' facial images are processed using the face_recognition library — built on top of Dlib's deep metric learning model — to generate unique 128-dimensional face embedding vectors that mathematically encode each individual's facial geometry. At recognition time, the system captures a live webcam feed, detects all faces in each frame, generates embeddings for each detected face, and computes cosine distances against the enrolled embedding database stored in SQLite. When a match is confirmed within a configurable threshold, the system logs the student or employee name, date, and precise timestamp into the SQLite attendance database, avoiding duplicate entries through a session-aware recording mechanism. Daily and monthly attendance reports are exportable as CSV files. This system is a real-world applicable project demonstrating biometric security, deep learning-based face analysis, and database-driven record management.

 **Difficulty: Advanced** **Suitable for: B.E. / B.Tech / MCA****SYSTEM DATA FLOW / ARCHITECTURE**Webcam Feed
FrameFace Location
Landmarks128-d Face
EmbeddingsCosine
Distance MatchLog SQLite DB
Time

Get in Touch with SRUA

Have a project in mind? Need a custom recommendation? Our founders respond within hours. Reach out directly on WhatsApp for the fastest support — we are here to help you find the right project, understand the codebase, and deliver a successful submission.

FOUNDER

Rohan Kokatare

Primary point of contact for all project inquiries, source code queries, and technical setup support.

 kokatarerohan@gmail.com

 **+91 99673 59705**

CO-FOUNDER

Saurabh

Available for documentation support, report queries, PPT assistance, and project delivery coordination.

 saurabhy.in@gmail.com

 **+91 86556 47416**