Project Ideas for Class 11 & 12 Students

16 team projects students can build with the help of AI tools: 8 AI-based and 8 non-AI

About this handout

Every project below solves a real problem that students can see around them in their own town. Each one is designed for a team of 2-4 students working for 2-4 weeks on a school PC with internet access. Students are expected to know basic Python, HTML and CSS and to use AI tools (ChatGPT, Gemini, Claude, etc.) as a tutor and coding helper. All tools listed are free.

Each project lists the Idea / Problem Definition (with features the team can include), the Tools Required, and the Learning Outcome. Choosing a project that matches their interest also helps students explore a possible career area, as shown in the table below.

Projects at a glance

No.	Project	Type	Career areas it points to
1	AI Career Guidance Assistant	AI	Counselling tech, AI developer
2	Smart Attendance System with Face Recognition	AI	AI / computer vision engineer
3	Syllabus Doubt-Solver Chatbot	AI	AI / NLP developer, EdTech
4	Plant Disease Detector for Local Farmers	AI	AgriTech, machine learning
5	AI Notes Summariser and Quiz Maker	AI	EdTech, content and AI tools
6	Spam and Scam Message Detector	AI	Cyber security, data science
7	AI Interview and Communication Coach	AI	HR, communication, AI developer
8	Text Reader for the Visually Impaired (OCR + Voice)	AI	Assistive technology engineer
9	School Notice and Events Portal	Non-AI	Web developer
10	Pocket-Money and Expense Tracker	Non-AI	Commerce, CA, fintech
11	Library Management System	Non-AI	Software / database developer
12	Digital Catalogue and WhatsApp Order Page for a Local Shop	Non-AI	Entrepreneurship, digital marketing
13	Online Quiz and Mock Test App	Non-AI	EdTech, web developer
14	Blood Donor and Emergency Contact Directory	Non-AI	Healthcare tech, social work
15	Student Result Analyser Dashboard	Non-AI	Data analyst
16	Government Schemes and Scholarship Finder	Non-AI	Public policy, civil services, web

Section A: AI-Based Projects (1-8)

These projects use an AI model, either through a free API, a no-code training tool, or a Python library, as the core of the solution.

Project 1: AI Career Guidance Assistant	
Idea / Problem Definition	Students in smaller cities often have no access to a career counsellor and only know a handful of options. Build a web app where a student answers 10-15 questions about interests, favourite subjects, strengths and work style. The answers are sent to an AI model, which suggests 3-5 suitable careers, the subjects and entrance exams needed after Class 12, and free learning resources. **Can include:** • Interest quiz with scoring • Stream selector (Science / Commerce / Arts) • "Why this career fits you" explanation • Downloadable result card
Tools Required (all free)	• HTML, CSS, JavaScript for the quiz form • Python with Streamlit (or Flask) • Free-tier AI API (e.g., Google Gemini) or a free AI chatbot for prompt design • GitHub / Streamlit Community Cloud for free hosting
Learning Outcome	• Prompt engineering: getting structured answers from AI • Connecting a front end to an AI API and handling JSON • How recommendation systems work • Awareness of career paths beyond the usual options

Project 2: Smart Attendance System with Face Recognition	
Idea / Problem Definition	Manual roll call wastes 5-10 minutes per period and proxy attendance is common. Build a system that registers student faces through a webcam, recognises them in a live feed or class photo, marks attendance automatically in a spreadsheet and produces a monthly report. Get written consent from students and teachers, and keep all data on the school PC only. **Can include:** • Student registration screen • Daily and monthly CSV / Excel report • Low-attendance list • Admin password protection
Tools Required (all free)	• Python, OpenCV, face_recognition (or DeepFace) • pandas with CSV or SQLite for records • Tkinter or Streamlit for the interface • Webcam; Google Colab for testing
Learning Outcome	• Computer vision basics: face detection vs recognition • Working with image data, files and databases in Python • Building a simple GUI • Data privacy and ethics of using face data

Project 3: Syllabus Doubt-Solver Chatbot	
Idea / Problem Definition	Students in small towns cannot always get help after school or coaching hours. Create a chatbot that answers questions only from the student's own textbook chapters (for example NCERT Science, Maths or Accountancy), explains in simple English with a worked example, and says "not found in syllabus" when it is unsure. **Can include:** • Subject and chapter selector • "Explain in simpler words" mode • Practice-question button • Chat history
Tools Required (all free)	• Python and Streamlit for the chat interface • Free-tier AI API (e.g., Gemini) or an open-source model via Hugging Face • PyPDF to read textbook PDFs • Simple keyword search or sentence-transformers to pick the right passage
Learning Outcome	• How chatbots and large language models work • Feeding trusted text to AI to reduce wrong answers (hallucination) • Text processing and building a chat UI • Checking AI answers critically against the textbook

Project 4: Plant Disease Detector for Local Farmers	
Idea / Problem Definition	Many families in and around tier-3 towns depend on farming, and crop diseases are often spotted late. Students collect or download leaf photos (healthy vs diseased, e.g., tomato, potato, wheat), train an image classifier, and build a web page where a user uploads a leaf photo and gets the likely disease, a confidence score and basic care advice. **Can include:** • Confidence percentage with result • Prevention and treatment tips • Support for 2-3 crops • Nearby Krishi Vigyan Kendra / agriculture helpline details
Tools Required (all free)	• Google Teachable Machine (no-code training) with TensorFlow.js • Or Python + TensorFlow/Keras in Google Colab • PlantVillage dataset (Kaggle) plus own photos • HTML, CSS, JavaScript; GitHub Pages for hosting
Learning Outcome	• Supervised learning: data, labels, training and testing • Accuracy and overfitting, tested with new photos • Deploying a trained model inside a web page • Applying technology to agriculture

Project 5: AI Notes Summariser and Quiz Maker	
Idea / Problem Definition	Revising long chapters before exams takes a lot of time. Build a tool where a student pastes or uploads a chapter and receives a one-page summary, key terms, flashcards and 10 MCQs with answers. Teachers can use the same tool to prepare practice papers. **Can include:** • Difficulty level choice • Export summary and quiz to PDF • Score tracking • "Weak topics" list
Tools Required (all free)	• Python and Streamlit • Free-tier AI API (e.g., Gemini) • PyPDF / pdfplumber to read files • fpdf or reportlab for PDF export; SQLite for score tracking
Learning Outcome	• Using AI for summarisation and question generation • Getting structured (JSON) output through prompts • File handling and PDF generation • Judging the quality of AI-generated content

Project 6: Spam and Scam Message Detector	
Idea / Problem Definition	Fake lottery messages, job scams and false "government scheme" forwards on SMS and WhatsApp cheat many families. Train a model to classify a message as scam or genuine, and build a simple page where a user pastes a message and gets a verdict along with the reasons (urgency, suspicious link, request for money or OTP). **Can include:** • Local scam examples added to the dataset • "Why this was flagged" explanation • Safety tips page • Cyber-crime reporting information (helpline 1930, cybercrime.gov.in)
Tools Required (all free)	• Python, pandas, scikit-learn (TF-IDF + Naive Bayes) • Google Colab for training • Public SMS spam dataset (Kaggle / UCI) • Streamlit or Flask for the web page
Learning Outcome	• Complete machine-learning workflow: clean, train, test • Text classification; accuracy and precision • Why local examples improve a model • Cyber-safety awareness they can share with their families

Project 7: AI Interview and Communication Coach	
Idea / Problem Definition	Students from smaller towns often feel nervous about admission interviews, job interviews and English speaking. Build a practice tool that asks interview questions for a chosen course or job, accepts a typed or spoken answer, and gives AI feedback on clarity, grammar, confidence and how to improve, with a score that is tracked over sessions. **Can include:** • Question bank by field (engineering, nursing, banking, teaching, etc.) • Voice input and spoken questions • Feedback checklist • Progress chart across sessions
Tools Required (all free)	• Python and Streamlit • Free-tier AI API (e.g., Gemini) • SpeechRecognition (or browser speech API) and gTTS • matplotlib for the progress chart
Learning Outcome	• Writing role-based prompts with a scoring rubric • Speech-to-text and text-to-speech • Working with APIs and plotting data • Interview and communication skills

Project 8: Text Reader for the Visually Impaired (OCR + Voice)	
Idea / Problem Definition	Visually impaired and low-literacy users often cannot read printed notices, bills or books. Build an app where a photo of printed text is converted into digital text (OCR) and read aloud, with speed control and the option to save the text. An AI model can optionally shorten long text into a simple summary. **Can include:** • Large-font, high-contrast mode • Adjustable reading speed • Keyboard-only navigation • Optional AI summary of long text
Tools Required (all free)	• Python, OpenCV, pytesseract (Tesseract OCR) • gTTS or pyttsx3 for speech • Tkinter or Streamlit for the interface • Free-tier AI API for summaries (optional)
Learning Outcome	• OCR and image pre-processing • Text-to-speech • Accessibility and inclusive design • Empathy-driven problem solving; careers in assistive technology

Section B: Non-AI Projects (9-16)

These projects build core programming, data and web skills and solve everyday local problems without needing an AI model. AI tools can still be used by students for learning, debugging and design help.

Project 9: School Notice and Events Portal	
Idea / Problem Definition	Notices are shared on paper or in scattered WhatsApp groups and are often missed. Build a website with a notice board, event calendar, timetable, exam datesheet, photo gallery and contact page. A teacher should be able to add a notice without touching the code. **Can include:** • Category filters and search • "New" badge for recent notices • PDF download links • Mobile-friendly layout
Tools Required (all free)	• HTML, CSS, JavaScript; VS Code • Google Sheets / Google Forms as a simple data source • GitHub Pages or Netlify for free hosting • AI chatbot for code help and design ideas
Learning Outcome	• Website structure and responsive design • Reading data into a page with JavaScript • Using Google Sheets as a lightweight database • Version control with Git and publishing a live site

Project 10: Pocket-Money and Expense Tracker	
Idea / Problem Definition	Students and families rarely track small expenses, so savings goals fail. Build an app to record income and expenses by category, show monthly totals, set a savings goal and warn when a category budget is crossed. **Can include:** • Category-wise pie chart • Monthly report download (CSV) • "Needs vs wants" tag • Data saved in the browser
Tools Required (all free)	• HTML, CSS, JavaScript with localStorage and Chart.js • Or Python with Tkinter, pandas and matplotlib • VS Code / IDLE
Learning Outcome	• Create, read, update and delete (CRUD) operations • Saving and loading data; arrays and objects • Data visualisation • Basic financial literacy; links to commerce, CA and fintech careers

Project 11: Library Management System	
Idea / Problem Definition	School library records are kept in registers, so it is hard to know who has which book. Build a system to add books and members, issue and return books, calculate fines for late returns and search the catalogue. **Can include:** • Overdue books list • Most-borrowed books report • Member ID card number entry • Admin login
Tools Required (all free)	• Python with SQLite • Tkinter (desktop) or Flask with HTML/CSS (web) • DB Browser for SQLite • AI chatbot for debugging help
Learning Outcome	• Database design: tables, keys and relationships • Writing SQL queries • Object-oriented thinking and form design • Analysing a real-world system before building it

Project 12: Digital Catalogue and WhatsApp Order Page for a Local Shop	
Idea / Problem Definition	Small shops (kirana, tailor, sweets, mobile accessories) lose customers who cannot see what is available. With the owner's permission, build a mobile-friendly catalogue with photos, prices and categories, plus an "Order on WhatsApp" button that sends a pre-filled message with the chosen items. **Can include:** • Cart with running total • Shop timing and map link • QR code for the shop counter • Offers section
Tools Required (all free)	• HTML, CSS, JavaScript • WhatsApp click-to-chat link (wa.me) and Google Maps embed • Free QR code generator; Canva (free) for images • GitHub Pages or Netlify
Learning Outcome	• Front-end development and cart logic • URL encoding and building dynamic links • Gathering requirements from a real client • Entrepreneurship and digital marketing basics

Project 13: Online Quiz and Mock Test App	
Idea / Problem Definition	Practice tests for school or entrance exams are usually on paper and checked by hand. Build a timed quiz app with a question bank by subject, random question order, instant score, answer review and result history. **Can include:** • Countdown timer • Negative-marking option • Class-wise leaderboard • Topic-wise performance analysis
Tools Required (all free)	• HTML, CSS, JavaScript • Questions stored in a JSON file • localStorage for results (or Flask + SQLite for many users) • GitHub Pages
Learning Outcome	• JavaScript logic: loops, conditions, timers • Working with JSON and randomisation • Score calculation and feedback design • Understanding exam patterns and performance analysis

Project 14: Blood Donor and Emergency Contact Directory	
Idea / Problem Definition	In an emergency, finding a blood donor, ambulance, oxygen supplier or open pharmacy in a small town depends on phone chains. Build a directory where volunteers register (blood group, area, availability) and anyone can search by blood group and locality, along with a page of verified emergency numbers and hospitals. **Can include:** • Hide donors who donated in the last 3 months • Click-to-call buttons • Verification by a teacher or local NGO • Privacy: show first name only until the request is confirmed
Tools Required (all free)	• HTML, CSS, JavaScript • Google Forms + Google Sheets as the backend (or Firebase free tier) • Google Maps links
Learning Outcome	• Collecting data with forms and filtering it • Privacy-by-design thinking • Verifying information before publishing it • Social responsibility through technology

Project 15: Student Result Analyser Dashboard	
Idea / Problem Definition	Teachers calculate class results by hand in Excel every term. Build a Python tool that reads marks from a CSV or Excel file, calculates totals, percentages, grades and ranks, finds subject-wise averages and weak students, and generates charts and printable report cards. **Can include:** • Topper list and pass/fail count • Improvement compared with the previous test • PDF report card for each student • Subject-wise difficulty chart
Tools Required (all free)	• Python, pandas, matplotlib, openpyxl • Streamlit or Tkinter for the dashboard • fpdf or reportlab for PDF report cards
Learning Outcome	• Data handling with pandas • Statistics: mean, median, percentage, rank • Charts and automation of repetitive work • Introduction to data analytics careers

Project 16: Government Schemes and Scholarship Finder	
Idea / Problem Definition	Many students and families in tier-3 towns do not know which schemes, scholarships or benefits they qualify for. Build a portal where a user selects category, income range, class or course, state and gender, and a rule-based filter shows matching schemes with eligibility, documents needed and the official link. **Can include:** • Document checklist for each scheme • Last dates and deadlines • Printable summary • Links to official websites only (e.g., scholarships.gov.in)
Tools Required (all free)	• HTML, CSS, JavaScript • Scheme data in JSON or Google Sheets, collected from official websites • GitHub Pages
Learning Outcome	• Research and verifying data from official sources • Rule-based logic (if-else filters) • Building searchable lists and writing clear public information • Civic awareness; links to public policy and civil services careers

Tips for Teams

- **Use AI as a tutor, not a shortcut.** Ask the AI to explain every piece of code it gives, and make sure every team member can explain their own part.
- **Keep a prompt diary.** Save the prompts that worked and the ones that failed. It makes a strong part of the final presentation.
- **Start small.** Build a working basic version in the first week, then add the "can include" features one by one.
- **Test with real users.** Ask classmates, teachers or family members to try the project and note what confuses them.
- **Protect personal data.** Take permission before using anyone's photo, phone number or marks, and never put real personal data in public pages or AI chats.
- **Present the problem first.** In the final demo, explain who has the problem, how the project solves it, and what the team learned.

For any help you can message me on WhatsApp

9009419666