IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155227

Strengthening CBSE Computer Science and Informatics Practices Education: A Proposed Framework for AI Integration, Global Awareness, and Career Readiness at Class XI–XII Level

Neerja Jain

Lecturer Computer Science, Department of Computer Science,

Government Aided Senior Secondary School, Delhi, India

Abstract: Globally, secondary school students are learning to train machine learning models, build AI-powered applications, and deploy code on cloud platforms — in countries as varied as Singapore, Finland, China, and the United States. In India, Computer Science (Code 083) and Informatics Practices (Code 065) provide a strong foundation in Python, Pandas, SQL, and networking — skills that are genuinely relevant and correctly chosen. Yet a student completing Class XII in either subject has not encountered Artificial Intelligence in any practical form within their main course, has no awareness that languages such as R, Julia, JavaScript, and Rust drive significant portions of the global AI and data economy, and has received no guidance on where two years of technical learning leads professionally. This paper presents a systematic analysis of the CBSE Class XI–XII CS and IP syllabi, identifies three specific gaps — absent AI integration, narrow technology horizon, and invisible career pathways — and proposes a concrete, implementable framework to address each. The framework introduces AI concepts naturally within existing syllabus topics (including a three-level ML introduction requiring no new infrastructure), strengthens practical learning through a Build-Deploy-Share model, builds global language awareness, and maps subject skills to explicit career pathways for both science and commerce stream students. Every proposed change is designed for immediate adoption in government senior secondary schools without new subjects, new examinations, or additional cost.

Keywords: CBSE Computer Science, Informatics Practices, AI Integration, Future-Ready Education, Career Pathways, NEP 2020, Python, Global Programming Languages, Government Schools, Practical Learning

I. INTRODUCTION

Every year, thousands of students across India complete their Class XII Computer Science or Informatics Practices examination, submit their Python projects, and walk out of the examination hall having written correct SQL queries and file handling programs. They have, by every official measure, passed. But ask them six months later — can you build something independently? Can you analyse real data and draw a meaningful conclusion from it? Do you know what Artificial Intelligence actually does, and whether your skills connect to it in any way? More often than not, the answer is silence.

This is not a critique of students, or of the curriculum's foundations. It is an observation about a specific and addressable gap between what two years of CS or IP education currently delivers and what the rapidly evolving technology landscape now requires.

The author has taught Computer Science (Code 083) and Informatics Practices (Code 065) in a government senior secondary school in Delhi for over seventeen years. In that time, the curriculum has evolved meaningfully — the shift from C++ to Python was a significant and welcome step, and the introduction of data handling libraries like Pandas and Matplotlib in IP has genuinely modernised the subject for commerce stream students. These changes deserve acknowledgment. However, after seventeen years of sitting across from students in a computer lab, watching them type programs they understand syntactically but cannot connect to the world outside the classroom, a question has become impossible to ignore: are we preparing students for the technology landscape that actually exists, or for the one that existed when the syllabus was last substantially revised?

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

Artificial Intelligence is not a future technology. It is the present one. It shapes the search results students see every morning, decides what content plays next on their phones, and increasingly influences hiring decisions, medical diagnostics, and financial systems. The National Education Policy 2020 recognises this and calls explicitly for computational thinking, AI awareness, and practical project-based learning across school stages. Yet a student who studies CS or IP through Class XII encounters no formally assessed AI content within their main subject. AI exists in the CBSE framework as a separate, optional skill elective — one that fewer than six percent of government schools are in a position to offer. For the student in the regular CS or IP classroom, it is simply absent.

This paper does not argue that the current CBSE CS and IP syllabi are poorly designed — they are not. Python is the right foundational language. Data handling and SQL are genuinely useful skills. The project component, when implemented as intended, has real merit. What this paper argues is that the syllabi, strong in their foundations, carry three specific gaps that, if addressed thoughtfully, would transform them from competency-building curricula into future-ready ones.

The first gap is the complete absence of AI integration within CS and IP — not as a new subject or an overloaded addition, but as a natural lens applied to topics that already exist. The second gap is the narrowness of the technology horizon presented to students: Python and SQL are taught correctly as primary tools, but students graduate with no awareness of the broader language ecosystem that drives global AI and technology work. The third gap is the absence of career context — the syllabi build technical skills without once telling students where those skills lead.

This paper proposes a strengthened framework for CBSE Class XI–XII CS and IP that retains existing strengths, integrates AI naturally within the current structure, introduces global language awareness as a conceptual addition, and connects learning outcomes to explicit career pathways. The goal is not to replace what exists, but to make it do considerably more.

II. CURRENT SYLLABUS ANALYSIS: WHAT EXISTS AND WHAT IS MISSING

A. Computer Science (Code 083) — Class XI

The Class XI CS syllabus covers three units: (1) Computer organisation — hardware, Boolean logic, number systems, encoding; (2) Python programming (45 of 70 theory marks) — data types, control structures, strings, lists, tuples, dictionaries, standard library modules; (3) Society, law and ethics — cyber crime, IT Act, e-waste. The practical component builds foundational syntax through pattern generation, series computation, and string manipulation — exercises that are entirely appropriate for students encountering a programming language for the first time. Python entered the CBSE CS syllabus in 2019–20, replacing C++. The language choice is right and well-timed in the Indian curriculum context. What is absent from Class XI CS: any connection between the Python students are learning and its dominant use in Artificial Intelligence and data science; any awareness of the broader language ecosystem; any career context.

B. Computer Science (Code 083) — Class XII

Class XII CS builds on XI foundations: functions, file handling (text, binary, CSV), exception handling, stacks (data structure), computer networking, and relational database management with Python-SQL connectivity. The CSV file handling unit is particularly significant — CSV is the universal format for machine learning training datasets, yet this connection is never made in the current syllabus. Similarly, the statistics module (mean, median, standard deviation) introduced in Class XI mirrors precisely the first operations any ML pipeline performs on a training dataset. The Class XII project brief is well-intentioned — it asks students to find a real-world problem and build something useful — but without scaffolding tools such as GitHub for version control, Streamlit for deployment, or scikit-learn for ML-based projects, students default to file-handling or SQL applications. What is absent from Class XII CS: Machine Learning concepts, AI tool familiarity, deployment skills, global language awareness, and career guidance.

C. Informatics Practices (Code 065) — Class XI and XII

IP 065 is designed for commerce and non-mathematics students and takes a more applied approach. Class XI covers Python basics, SQL fundamentals, and — notably — an ‘Emerging Trends’ unit. This unit names Artificial Intelligence, Machine Learning, Natural Language Processing, Big Data, Blockchain, Cloud Computing, IoT, and more. On paper, it sounds comprehensive. In practice, the unit is entirely theoretical: students read about these technologies, understand their definitions for examination purposes, and move on. There is no hands-on component, no connection to the Python they have already learned, and no project expectation. It is, in effect, a chapter of vocabulary.
Class XII IP is where the subject genuinely comes into its own: Pandas and Matplotlib for data handling and visualisation, advanced SQL, and SQL-Pandas connectivity. These are real data skills. A student who learns Class XII

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

IP thoroughly has a working knowledge of data manipulation and visualisation that is directly applicable in data analytics careers. The proposal here is not to diminish what exists, but to take it one meaningful step further — connecting these skills explicitly to AI and to the professional roles they lead toward.

It is important to acknowledge that experienced CS and IP teachers already make real-world connections in the classroom — linking Python list operations to how data is stored in applications, or SQL queries to how apps retrieve information for users. These connections reflect the quality and initiative of individual teachers. The gap is not in teaching quality; it is in formal curricular recognition of these connections, so that every student — regardless of which school they attend or which teacher they have — receives them consistently and as part of the official learning journey. A further concern, less discussed in curriculum literature, is what happens to this learning after school. Students who do not continue with Computer Science in higher education find that their Python and SQL knowledge — genuinely useful skills — gradually fades without an active context to apply it in. However, the rise of accessible AI tools has created a new and significant opportunity: a student who understands basic data logic, what a query does, and how a function works — all of which CS and IP already teach — can use AI-powered tools like ChatGPT, Gemini, Google Sheets AI, or no-code automation platforms to simplify their own work and others' work, regardless of what career path they eventually pursue. This argues strongly for embedding AI tool literacy within CS and IP, alongside AI programming concepts, so that the knowledge students build in school remains applicable and alive long after they leave it.

D. Summary: The Three Gaps

TABLE I SUMMARY OF GAPS IN CS 083 AND IP 065 SYLLABI

Gap	CS 083	IP 065	Impact
AI Integration	Absent entirely	Named in XI, no depth or practical	Students have zero AI understanding from 2 years of study
Global Language Awareness	Python + SQL only	Python + SQL only	No awareness of R, JavaScript, Julia, Rust, Go
Career Pathways	Not mentioned	Not mentioned	Students cannot connect their learning to professional opportunity
Practical Depth	Pattern programs, palindromes	Data handling, some depth in XII	CS practical lags behind real-world relevance
Independence Skill	Not addressed	Not addressed	Students cannot build, deploy, or share anything independently

III. LITERATURE REVIEW

A. NEP 2020 and the Technology Mandate

The National Education Policy 2020 is explicit in ways that make the gaps in CS and IP syllabi all the more visible. Section 4.26 mandates coding and computational thinking from Grade 6, with the clear intention of building progressive digital fluency through secondary school. Section 16 integrates vocational education — including high-value technology skills — into the mainstream curriculum. The policy's overall emphasis on holistic, multidisciplinary, and project-based learning directly implies that technology education should connect to real problems and real outcomes, not merely to examination performance. The current CS and IP syllabi are not in conflict with NEP 2020 — but they are not yet fulfilling its spirit either.

B. AI Literacy in School Education

Research on AI literacy in school contexts has grown substantially in recent years. Long and Magerko (2020) define AI literacy as a set of competencies that allow individuals to understand, use, evaluate, and create with AI systems in contextually and socially appropriate ways. Ng et al. (2021) identify four dimensions of AI literacy: knowing and understanding AI, using and applying AI, evaluating and creating with AI, and engaging with AI ethics. What is significant about both frameworks is their emphasis on practical engagement — students do not become AI-literate by reading about AI; they become AI-literate by working with it, even in simple ways.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

Touretzky et al. (2019) proposed five foundational ideas for K–12 AI education: Perception, Representation and Reasoning, Learning, Natural Interaction, and Societal Impact. Several of these map naturally onto topics that already exist in the CBSE CS and IP syllabi. Data handling in Pandas is directly related to how AI systems represent and learn from information. The societal impact discussions in both subjects are the natural home for AI ethics. The connection is already latent in the syllabus — it simply has not been made explicit.

C. International Context

A brief survey of international secondary school CS curricula reveals a consistent pattern: AI and data science are integrated within mainstream computing subjects, not separated into optional electives. Table III presents Singapore's Computing curriculum as a detailed illustration, chosen because it is the most explicitly documented and closest in structure to India's board-based system.

TABLE II SINGAPORE O-LEVEL AND A-LEVEL COMPUTING: LANGUAGE AND AI PROGRESSION BY GRADE

Grade / Level	Subject	Languages Taught	AI / Data Content	Practical Outcome
Secondary 3–4 (~Class IX–X equivalent)	Computing (O-Level)	Python (primary); HTML/CSS for web	Intro to AI concepts; data representation; algorithmic thinking	Working Python programs; small web project; problem-solving portfolio
Secondary 3–4 (enrichment)	Applied Learning Programme (ALP)	Python; Scratch (earlier grades)	Machine learning basics using supervised learning examples; AI ethics discussion	Simple ML classifier using free tools; ethics case study presentation
Junior College 1–2 (~Class XI–XII equivalent)	H2 Computing (A-Level)	Python (advanced); SQL; exposure to R and JavaScript	Data structures for AI; algorithm analysis; neural network concepts; real dataset projects	End-to-end data project; deployed application; GitHub portfolio submission
Junior College (enrichment)	AI for Everyone / Digital Literacy	Python; no-code AI tools (Teachable Machine, ML for Kids)	Hands-on ML model training; NLP basics; AI societal impact	Working AI application; documented project with reflection on ethics and impact

Several observations from Singapore's curriculum are directly relevant to the CBSE context. First, Python is introduced early and deepened progressively — it is not restarted at each new level. Second, AI is not a separate subject: it is woven into the Computing curriculum at each stage, using the same Python environment students already know. Third, by the A-Level equivalent (Class XI–XII), students are using GitHub for version control and deploying working applications — skills that have zero cost and are accessible on standard hardware. Fourth, R and JavaScript are introduced conceptually at the senior level, not as additional programming subjects but as awareness of a broader ecosystem.

China's approach is more ambitious: from September 2025, AI has been mandated as a compulsory component across all school levels, integrated within existing Information Technology and General Technology courses. Hong Kong requires 10–14 hours of AI education annually for junior secondary students. The United States College Board's AP Computer Science Principles course, taken by over 200,000 students annually, includes a dedicated Create Performance Task in which students build and document an original program — assessed on functionality, not just syntax. The common thread across all these systems is the same: AI and applied programming are integrated into the core subject, not placed in a separate elective that only a fraction of students access.

D. Global Programming Language Landscape and Job Market

The 2025 Stack Overflow Developer Survey and LinkedIn Jobs Report together provide a clear picture of what languages the global AI and technology job market actually requires. Python leads with approximately 58 percent of AI

DOI: 10.17148/IJARCCE.2026.155227

and ML job postings globally and holds the top position in the TIOBE Index at approximately 26 percent share — confirming that CBSE's choice of Python as the primary language is correct. However, Python is not the whole story, and students who know only Python will encounter a narrower range of opportunities than those who understand the ecosystem around it.

TABLE III GLOBAL PROGRAMMING LANGUAGES IN AI AND TECHNOLOGY: ROLE, JOB DEMAND, AND SCHOOL CURRICULUM RELEVANCE

Language	Primary Use in AI / Tech	Global Job Demand (2025)	Who Uses It	What CBSE Students Should Know
Python	ML, data science, automation, scripting, AI research	Highest — 58% of AI job postings globally; TIOBE #1 at 26%	Google, Meta, ISRO, every data team globally	Primary language — already in syllabus; students should understand WHY it dominates AI
R	Statistical analysis, health research, biostatistics, academic data science	Strong in research and pharma; 11% of data science postings	WHO, pharmaceutical companies, universities, ICMR-type bodies	If interested in medicine, biology, public health, economics — R is the domain language
JavaScript / TypeScript	Browser-based AI (TensorFlow.js), web apps, full-stack development	Most-used language overall; 66–69% of all developers use it	Every web company; AI that runs without a server	AI can run in a browser with no GPU — directly relevant for low-infrastructure schools
Julia	Scientific computing, climate modelling, quantitative finance, research labs	Growing in research institutions; C-level speed with Python-like syntax	MIT, NASA, climate research groups, financial AI teams	For students interested in physics, mathematics, or research careers
Rust	AI infrastructure, systems programming, memory-safe software	Fastest-growing in systems roles; most-admired language 5 years running (Stack Overflow)	Microsoft, Amazon, Google — building AI infrastructure layers	The language that runs AI systems; not for beginners but important to know it exists
Go (Golang)	Cloud computing backends, scalable AI services, Kubernetes	High demand in cloud and DevOps roles	Google, Uber, Docker, cloud-native companies	Powers the cloud platforms where AI models are deployed and served
SQL	Data querying across all industries and AI pipelines	Required in virtually every data role globally	Every organisation with a database — universal	Already in syllabus — students should understand it as a lifelong skill, not just an exam topic

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

The curriculum implication is not to teach all these languages — that would be impractical and counterproductive. The proposal is a single 4–6 period Technology Landscape unit, discussion-based and requiring no programming, that covers this table conceptually. A student who leaves Class XII understanding Python's place in this ecosystem — and knowing that R, JavaScript-AI, Julia, and Rust exist and why — is equipped to make informed decisions about further learning and career direction. A student who knows only that Python and SQL exist is not.

IV. GAP ANALYSIS: THREE SPECIFIC FINDINGS

A. The AI Absence

The most significant gap in both CS 083 and IP 065 is the complete absence of Artificial Intelligence as a practical, integrated element of the main subject. The Class XI IP 'Emerging Trends' unit acknowledges that AI exists, but acknowledgment is not literacy. A student who can define machine learning for an examination question, but has never trained a simple model, explored a dataset with AI intent, or understood why the Pandas operations they already know are foundational to ML pipelines, has not gained AI literacy — they have gained AI vocabulary.

The problem is compounded by the positioning of AI as a separate elective (Code 843). This design sends an implicit message that AI is an optional extra for the particularly motivated, rather than a foundational literacy for all technology students. In government senior secondary schools, where resource and time constraints mean that optional subjects are often simply not offered, this separation effectively places AI outside the reach of the majority of students.

B. The Single-Language Horizon

Both CS and IP present students with a single-language world: Python for programming, SQL for data. This is not incorrect — Python and SQL are genuinely the right foundational tools, and the 2019–20 shift from C++ to Python was a significant and correct curriculum decision. The gap is the absence of any awareness that the world extends beyond them. Technology moves fast — Python had been the dominant language in global AI and data science for nearly a decade before it entered the CBSE syllabus. The curriculum cannot always move at the pace of industry, but it can equip students with the awareness to move at that pace themselves. A student who understands only that Python exists is not yet equipped to adapt; a student who understands Python's place in a larger, evolving ecosystem is.

Second, students who are unaware that R, Julia, Rust, or JavaScript-based AI tools exist cannot make informed choices about their own learning and career paths after Class XII. A student who knows that R is dominant in health data research, and who is interested in medicine or public health, has information to act on. Without that awareness, the connection is simply never made.

C. The Career Visibility Gap

Neither the CS nor the IP syllabus contains any explicit guidance on career pathways. This may appear to be beyond the scope of a subject syllabus — but it is worth considering what the absence means in practice, particularly for students in government schools where career counselling may be limited, where parents may not have technology sector experience, and where the connection between school subjects and professional opportunity is not self-evident.

A student who has spent two years learning Python, data handling, SQL, and networking has, without realising it, built the foundation for a career in data analysis, software development, database administration, cybersecurity, or AI-adjacent roles. A commerce stream student who has mastered Pandas, Matplotlib, and SQL has the core toolkit of a business analyst. Neither student is likely to know this unless someone tells them. The syllabus could be that someone.

V. PILLAR ONE: AI INTEGRATION WITHIN EXISTING SYLLABUS TOPICS

The first pillar of the proposed framework is AI integration — not as a new unit, not as additional examination content, but as a natural contextual layer applied to topics that already exist in both CS and IP. The principle is simple: every major topic in CS and IP touches something that AI also touches. Making that connection explicit costs no additional syllabus time and gives students a dramatically richer understanding of why they are learning what they are learning.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155227

A. Topic-Level AI Mapping — CS 083

TABLE IV AI INTEGRATION MAPPING FOR CS 083

Existing CS Topic	AI Connection to Add	What Student Understands
Python data types and lists (XI)	A dataset is a structured list; ML models learn from structured data	My Python lists are the same format AI uses to learn
Dictionaries and data structures (XI)	Key-value structures are how AI models store learned associations	Dictionary logic underlies how AI stores knowledge
Statistics module — mean, median, mode (XI)	Descriptive statistics are the first step in any ML data analysis	Statistics is not abstract; it is AI's first question about data
File handling — CSV files (XII)	CSV is the universal format for ML training datasets	The CSV files I write are exactly what I would feed a model
Stacks — data structure (XII)	Neural networks use stack-like forward and backward propagation logic	Data structures are the bones of AI computation
Computer Networks (XII)	AI-powered intrusion detection and network anomaly detection	Networks + AI = cybersecurity, a high-demand real-world field
Society, Law and Ethics (XI)	AI ethics: bias in algorithms, surveillance, data privacy, deepfakes	Ethics of AI is already part of my syllabus — I just need to name it
Project (XII)	Allow AI-based projects: simple classifiers, data analysers, chatbots using free tools	I can build something AI-related within my existing skills

B. Topic-Level AI Mapping — IP 065

TABLE V AI INTEGRATION MAPPING FOR IP 065

Existing IP Topic	AI Connection to Add	What Student Understands
Emerging Trends unit (XI)	Add hands-on component: explore Teachable Machine, discuss one real AI application in depth	AI is not just a definition; it is something I can try today
Pandas Series and DataFrames (XII)	DataFrames are how data scientists and ML engineers prepare training data	My Pandas work is directly used in AI pipelines
Data Visualisation — Matplotlib (XII)	Data visualisation is how AI teams communicate model findings and patterns	Charts I create are the same kind analysts use in AI projects
SQL joins and aggregate functions (XII)	Database queries are how AI applications retrieve training data at scale	My SQL skills connect directly to how AI systems access information
Societal Impacts unit (XII)	Add: AI and employment, algorithmic decision-making, AI tools in business	AI's impact on my future career is something I can think critically about
Project (XII)	Encourage data analysis projects with a predictive or pattern-finding element	My project can have genuine analytical value, not just data entry

DOI: 10.17148/IJARCCE.2026.155227

C. AI Tool Literacy — A Practical Addition for Both Subjects

Beyond AI concepts and programming, there is a third dimension of AI literacy that the current syllabus does not address: the ability to use existing AI tools productively. This is not a coding skill — it is a practical digital skill that remains useful regardless of career path. Proposed addition: one dedicated practical session per term (approximately 4–6 sessions across Class XI–XII) in which students use a specific AI tool to solve a defined problem. No examination marks are attached. These sessions extend naturally from skills students already have: a student who knows what a CSV file is can use Google Colab; a student who knows what a function does can use GitHub Copilot; a student who understands data can use Google Sheets AI.

TABLE VI AI TOOLS FOR CLASSROOM USE: ZERO COST, EXISTING INFRASTRUCTURE

AI Tool	What It Does	How CS / IP Students Can Use It	Infrastructure Needed
ChatGPT / Google Gemini	Text generation, explanation, code assistance, summarisation	Debug Python code, explain SQL errors, research project topics, draft README files	Browser only; free tier sufficient
Google Sheets AI (Gemini in Sheets)	Data analysis and insight generation using natural language queries	Analyse class datasets, generate charts, ask questions about data without writing formulas	Google account; browser only
GitHub Copilot (free for students)	AI-powered code completion and suggestion within the code editor	Write Python programs faster; understand alternative approaches; reduce syntax errors	GitHub student account; free
Google Colab	Free cloud-based Python notebook with GPU access	Run data science and ML code without local hardware constraints; share work as a live notebook	Google account; browser only; no download
Teachable Machine (Google)	Train image, audio, and pose classifiers in the browser without coding	Build a working AI model in one class period; understand ML intuitively before theory	Browser only; no GPU; works on existing school computers
Make.com / similar (free tier)	No-code workflow automation connecting apps and data sources	Automate repetitive tasks; understand how real business processes use data and logic	Browser only; free tier available
Canva AI / similar	AI-assisted design and content creation	Create project presentations, infographics, and visual documentation of technical work	Browser only; free tier available

These tools require no new infrastructure, no additional examination content, and no change to the theory syllabus. They can be introduced as part of the practical sessions, as demonstration activities, or as optional enrichment for project work. Their inclusion sends a clear and important message to students: the skills you are building here connect directly to tools that professionals, entrepreneurs, and independent learners use every day. That connection is motivating in a way that abstract syntax exercises, however necessary, are not.

D. Practical AI Tools — Zero Cost, Existing Infrastructure

A concern that immediately arises in any AI curriculum proposal for government schools is infrastructure. The tools proposed here are deliberately chosen to work within the constraints of standard government school computer labs: existing hardware, limited internet, no GPU requirement.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155227

TABLE VII FREE AI TOOLS SUITABLE FOR GOVERNMENT SCHOOL INFRASTRUCTURE

Tool	What It Does	Infrastructure Needed	Used For
Google Teachable Machine	Train image/audio/pose classifiers in the browser	Any browser, no GPU	Hands-on ML without coding
Anaconda Python (offline)	Full data science stack offline	Download once, no internet needed	Pandas, scikit-learn, Matplotlib
scikit-learn (via Anaconda)	Simple ML models in Python	Existing school computers	Extend Class XII data work into ML
MIT App Inventor (ML extension)	Build mobile AI apps without coding	Browser-based	Applied AI for commerce students
TensorFlow.js	Run AI models in the browser	Any browser	Demonstrate AI without server

VI. PILLAR TWO: PRACTICAL LEARNING — BUILD, DEPLOY, SHARE

The second pillar addresses a gap that is not unique to CBSE — it is common to examination-driven curricula everywhere: the disconnect between knowing how to write code and knowing how to build something with it. A student who can write correct Python programs for twenty prescribed exercises in their report file, and then correctly answer questions about those programs in a board examination, has demonstrated technical compliance. They have not necessarily demonstrated the ability to identify a problem, design a solution, implement it, and put it in front of someone who can use it. That is a different skill, and it is the one that employers, higher education institutions, and entrepreneurial endeavours actually require.

A. The Build-Deploy-Share Model

This paper proposes that both CS and IP syllabi incorporate a 'Build-Deploy-Share' competency as a structured component of the Class XII project. The model has three stages, each corresponding to a skill that is currently absent from the syllabus.

TABLE VIII BUILD-DEPLOY-SHARE MODEL: STAGES, ACTIVITIES, AND TOOLS

Stage	What It Means	How It Looks in Practice	Tools
Build	Create a working application that solves a real problem	Student identifies a real need in their school, home, or community and writes a Python application or data analysis that addresses it	Python, Pandas, scikit-learn, SQL
Deploy	Make it accessible to someone other than the creator	Student publishes their project on GitHub or deploys a simple web interface using Streamlit — both free and functional on modest hardware	GitHub (free), Streamlit (free)
Share	Explain what was built and why it matters	Student writes a README or gives a short presentation: problem, solution, how it works, what they learned	GitHub README, classroom presentation

This three-stage model does not require more examination time. It reframes what the existing project is asking students to do. The current project brief already says: find a real-world problem. The Build-Deploy-Share model gives students the scaffolding and tools to actually do that, rather than defaulting to a pre-existing template.

DOI: 10.17148/IJARCCE.2026.155227

B. Sample Projects Achievable within Government School Constraints

TABLE IX SAMPLE AI PROJECTS FOR CLASS XII CS AND IP

Project Idea	Subject	Skills Used	AI Element
School fee defaulter predictor	CS / IP	Python, Pandas, simple classification	Decision tree or logistic regression using scikit-learn
Delhi air quality trend analyser	IP	Pandas, Matplotlib, open government data	Pattern recognition and data-driven insight
Classroom attendance tracker with anomaly alert	CS	Python, CSV file handling, basic statistics	Flag unusual patterns using mean/std deviation
Sentiment analysis of student feedback	CS	Python, string processing, simple NLP	TextBlob or VADER — both free, offline-capable
Vegetable price prediction (mandi data)	IP	Pandas, Matplotlib, simple regression	Linear regression on freely available AGMARKNET data
Library book recommendation system	CS	Python, dictionary, data structures	Collaborative filtering concept, implemented simply

Every project above can be completed using free tools, on standard school computers, within the existing practical time allocation. None requires internet access beyond the initial download of tools. All of them are more meaningful, more useful, and more memorable than a pre-typed file handling program.

C. Proposed: Practical-Only Assessment Topics

One structural change that requires no revision of the theory syllabus or the written examination is the introduction of practical-only assessment topics — skills assessed exclusively in the practical examination, carrying no theory examination burden whatsoever. This approach preserves the existing written examination structure completely while creating space for genuinely applied skills to be formally recognised. The following topics are proposed as optional practical assessment tracks, offered alongside the existing practical work.

TABLE X PROPOSED PRACTICAL-ONLY ASSESSMENT TOPICS FOR CLASS XII

Proposed Practical-Only Topic	Subject	What the Student Does	Assessment Method
GitHub portfolio — project upload and README	CS + IP	Create a GitHub repository, upload project code, write a plain-language README explaining the project and its purpose	Examiner verifies live repository; viva on project decisions
Deploy a working app using Streamlit	CS	Deploy a functional Python application online using Streamlit's free hosting — accessible via a public URL	Examiner accesses the live URL during practical; student explains deployment process
Use an AI tool to analyse a dataset	IP	Use Google Colab, Google Sheets AI, or ChatGPT to analyse a provided dataset and produce a structured output or report	Output screenshot + viva on what the tool did and why the student chose their approach
Train and demonstrate a Teachable Machine model	CS + IP	Train an image or audio classifier using Google Teachable Machine, test it	Live demonstration to examiner; student explains what the model learned

Proposed Practical-Only Topic	Subject	What the Student Does	Assessment Method
		with new inputs, and explain the model's behaviour	and where it failed
Create an interactive data dashboard	IP	Build a multi-chart Matplotlib or Pandas dashboard on a real dataset, with a short written interpretation of the findings	Practical output reviewed; viva on data choices and interpretation

These topics ask students to do what professionals actually do: build something, put it somewhere, and explain it to someone. They require no new theory content, no new textbooks, and no additional written examination burden. They require only that the practical examination rubric be expanded to recognise them as valid assessment pathways — a change well within CBSE's existing assessment architecture.

VII. PILLAR THREE: GLOBAL LANGUAGE AWARENESS

The third pillar is perhaps the most conceptually different from what currently exists in the syllabus: introducing students to the global programming language landscape. To be clear, this is not a proposal to teach multiple programming languages within Class XI–XII. That would be impractical, counterproductive, and beyond the scope of what can reasonably be asked of teachers or students within existing time constraints. The proposal is more modest and, arguably, more valuable: to teach students how to think about programming languages, so that they understand Python's place in a larger ecosystem and are equipped to make informed choices as that ecosystem evolves.

A. A One-Unit Technology Landscape Overview

The proposal is to add a single unit — approximately four to six teaching periods — titled something like 'The Technology Landscape and Your Skills' to the Class XII syllabus of both CS and IP. This unit would be primarily conceptual and discussion-based, requiring no additional programming work. It would cover the following.

TABLE XI GLOBAL AI PROGRAMMING LANGUAGE LANDSCAPE AND CURRICULAR RELEVANCE

Language / Tool	Where It Is Used Globally	Why Students Should Know It Exists	Difficulty vs Python
Python	AI/ML research, data science, automation, scripting — dominant globally	Primary tool; students already learn this — understand its strengths	Baseline
R	Health research, biostatistics, pharmaceutical data science, academic research	Students interested in biology, medicine, public health — R is the gateway	Similar to Python in concept
JavaScript / TypeScript	Web development, browser-based AI (TensorFlow.js), full-stack applications	AI that runs without a server; the most accessible deployment environment	Different paradigm; approachable
Julia	Scientific computing, climate modelling, quantitative finance, research labs	For students interested in physics, economics, data-intensive research	Python-like syntax, C-like speed
Rust	AI infrastructure, systems programming, memory-safe software at Microsoft/Amazon/Google	The infrastructure language of the AI era; high-salary niche	Steeper learning curve

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

Language / Tool	Where It Is Used Globally	Why Students Should Know It Exists	Difficulty vs Python
Go (Golang)	Cloud computing, backend AI services, scalable applications	Powers the cloud platforms that run AI; growing fast in enterprise	Clean syntax, different model
SQL	Universal data query language across all industries	Already in syllabus; emphasise that SQL is not optional in any data role	Already learned

The teaching method for this unit does not require any hands-on programming. Teachers can use discussion, case studies, and simple visual comparisons. A question like 'why does a hospital data analyst use R instead of Python for their work?' generates thinking about data domains, professional contexts, and tool selection that is far more valuable than being able to recite language syntax. The goal is for students to leave Class XII understanding that they have a strong foundation in Python and SQL, that the technology world has many rooms, and that their foundation gives them the tools to walk into any of those rooms with further learning.

B. Language Awareness and Adaptability

There is a broader pedagogical argument here. The CBSE CS syllabus shifted from C++ to Python roughly a decade after Python became dominant in the industry. The students who had learned C++ were not wrong to learn it — but they were not equipped to adapt easily because their learning had been language-specific, not language-agnostic. A student who understands why Python is the right tool for data science, but also understands that Rust is the right tool for AI systems infrastructure, and that R is the right tool for health analytics, is a student who can look at a new technology problem in 2030 or 2035 and ask the right question: what is the right tool for this, and how do I learn it? That question is more durable than any specific language.

VIII. PILLAR FOUR: CAREER PATHWAYS — MAKING THE CONNECTION VISIBLE

The fourth pillar is in some ways the simplest to implement and perhaps the most immediately impactful for students in government schools: making career pathways explicit within the syllabus. This does not require new content. It requires a change in framing — connecting the skills students are already building to the professional opportunities those skills unlock.

A. Career Pathways for CS 083 Students (Science Stream)

TABLE XII CAREER PATHWAYS FOR CS 083 (SCIENCE STREAM) STUDENTS

Career Path	What They Do	CS Skills Used	Additional Learning Needed
Data Scientist / ML Engineer	Build and deploy predictive models for business and research	Python, data structures, algorithms, project	scikit-learn, statistics, domain knowledge
Software Developer	Build applications for web, mobile, enterprise	Python, data structures, file handling, SQL	Frameworks (Django, Flask), version control
Database Administrator	Design and manage databases for organisations	SQL, Python-SQL connectivity, file handling	Advanced SQL, database design, cloud platforms
Cybersecurity Analyst	Protect systems from attack, investigate breaches	Networks, protocols, society and ethics unit	Security tools, ethical hacking foundations
AI / Research Engineer	Develop AI systems and algorithms	Python, data structures, algorithms, maths	Deep learning frameworks, research methodology

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

Career Path	What They Do	CS Skills Used	Additional Learning Needed
Freelance Developer	Build software solutions for clients independently	Python, SQL, project skills	One specialist skill + GitHub portfolio + marketing
Government Technology Roles	NIC, CDAC, ISRO, public sector IT	Full CS foundation	Competitive exam preparation

B. Career Pathways for IP 065 Students (Commerce Stream)

TABLE XIII CAREER PATHWAYS FOR IP 065 (COMMERCE STREAM) STUDENTS

Career Path	What They Do	IP Skills Used	Additional Learning Needed
Business / Data Analyst	Analyse business data to support decision-making	Pandas, Matplotlib, SQL, data visualisation	Excel advanced, Power BI, business domain
Financial Data Professional	Work with markets, banking, and financial data	Pandas, statistics, SQL, data handling	Finance domain, R or Python advanced
Digital Marketing Analyst	Analyse campaign data, customer behaviour, ROI	Data visualisation, SQL, emerging trends	Google Analytics, social media data tools
ERP / Business Systems	Implement and manage enterprise software (SAP, Oracle)	SQL, database concepts, networking	ERP certification, business process knowledge
FinTech / InsurTech Analyst	Work at intersection of finance and technology	Pandas, SQL, Python basics, data handling	Financial regulations, API basics
Entrepreneurship	Build data-driven business or analytics consultancy	Full IP toolkit	Domain expertise + freelance platforms

C. Common Pathways for Both Streams

TABLE XIV COMMON CAREER PATHWAYS FOR CS AND IP GRADUATES

Pathway	Applicable To	Why Both Streams Qualify
EdTech / Online Teaching	CS + IP graduates	Python and data skills are in high demand as teaching content; YouTube, Udemy, NPTEL opportunities
Content Creation (Tech)	CS + IP graduates	Technical knowledge + communication skills = creator in a high-demand niche
Freelancing on global platforms	CS + IP graduates	Python data work and SQL are marketable on Upwork, Fiverr, Toptal today
Higher Education in CS/IT	CS + IP graduates	Both subjects qualify for BCA, B.Sc. CS, and related programmes

This career mapping does not need to appear as a separate examination unit. It can be woven into the introduction to each major topic: 'You are learning Pandas today. Here is what a data analyst does with Pandas every morning.' That sentence takes five seconds to say. Over two years, those five-second connections accumulate into a student who understands where they are going.

DOI: 10.17148/IJARCCE.2026.155227

IX. THE PROPOSED STRENGTHENED FRAMEWORK: COMBINED VIEW

The four pillars described in Sections V through VIII are not independent interventions. They are complementary and mutually reinforcing. AI integration makes the existing syllabus more relevant. Practical learning gives students the means to apply it. Global language awareness ensures they can adapt beyond what they have learned. Career pathway visibility ensures they know why it matters. Together, they constitute a framework that strengthens CS 083 and IP 065 without replacing them — building on solid foundations rather than starting again.

TABLE XV ALIF: STRENGTHENED CS AND IP FRAMEWORK — COMBINED COMPONENT VIEW

Framework Component	What Changes in Syllabus	Where It Fits	Implementation Effort
AI Concept Integration	Add AI connection notes to 6–8 existing topics in each subject	Within existing units; no new units required	Low — teacher awareness + one teacher training session
AI Ethics Expansion	Expand Society and Ethics unit to explicitly include AI ethics	Existing Unit 3 (CS) and Societal Impacts unit (IP)	Low — existing content; add 2–3 AI-specific examples
Practical AI Component	Allow AI-based project options in the Class XII project component	Existing project structure	Low to Medium — teacher familiarity with 1–2 free AI tools
Build-Deploy-Share	Add GitHub portfolio and Streamlit deployment as optional project pathway	Class XII project brief	Medium — requires one-time teacher upskilling
Technology Landscape Unit	Add 4–6 period conceptual unit on global language ecosystem	Class XII — new lightweight unit	Low — discussion-based, no new programming
Career Pathway Contextualisation	Add career connections to topic introductions in textbooks/support material	All units; no marks change needed	Very Low — textbook revision, teacher awareness
IP Emerging Trends Enhancement	Add hands-on component to Class XI IP Emerging Trends unit	Existing unit; practical component added	Medium — Teachable Machine or equivalent; browser-based

A. Specific Proposals: What to Add, What to Trim

The following table offers concrete, specific proposals at the topic level — not as prescriptions, but as a starting point for curriculum committee discussion. Some existing content can be slightly trimmed to create space for additions without increasing the overall syllabus load.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

TABLE XVI SPECIFIC PROPOSALS: TOPICS TO ADD, TRIM, OR ENHANCE

Current Topic	Proposed Action	Reason
ARPANET history and evolution (CS XII Networks)	Trim to 2–3 lines of context; reduce detailed coverage	Historical context useful; extended coverage has low relevance to student outcomes in 2025
Number system conversions — extensive drill (CS XI)	Retain concept and one method; reduce repetitive conversion exercises	Foundational concept is important; extended calculation drill is lower priority than application-based work
Emerging Trends chapter (IP XI) — theory only	Retain all content; add a 2-hour hands-on session using Teachable Machine or Google Colab	Chapter exists and is well-structured; it simply needs a practical component to move from vocabulary to experience
Society and Ethics unit (CS XI)	Expand to include AI ethics explicitly: algorithmic bias, deepfakes, AI and employment	Natural home for AI ethics; content is already present; AI dimension is absent
Class XII project brief (CS and IP)	Add Build-Deploy-Share as an explicitly described and encouraged option, naming GitHub and Streamlit	Project brief is well-written; scaffolding upgrade enables students to fulfil its stated intent
NEW — Technology Landscape unit (CS + IP, Class XII)	Add 4–6 periods; no examination marks; discussion-based	Equips students with language-agnostic thinking; prepares them for a dynamic technology ecosystem
NEW — AI Tool Literacy (CS + IP, Class XI or XII)	Add to practical sessions; no theory marks; assessed in practical only if adopted as optional track	Zero infrastructure needed; connects school learning to real professional and personal use
NEW — Practical-only assessment options (CS + IP, Class XII)	Add optional assessment tracks: GitHub portfolio, Streamlit deployment, Teachable Machine demonstration	No theory burden; creates formal recognition for applied skills that employers and higher education value

X. DISCUSSION

The framework proposed in this paper rests on a fundamental observation: the gap between what CS and IP currently offer and what they could offer is not primarily a resource gap. It is a framing gap. The tools are free. The infrastructure is already in place. The Python knowledge students build in these subjects is genuinely relevant to AI, data science, and professional technology work. The problem is that the current syllabus does not make these connections — and in government schools, where the syllabus is the curriculum, connections that are not made in official documents are connections that are not made at all.

This matters most for students in government schools. A student in a well-resourced private school has teachers who supplement the syllabus, parents who can identify career opportunities, coaching institutes that cover topics beyond the board curriculum, and exposure to a professional network that can show them what a data scientist or ML engineer actually does. A student in a government senior secondary school may have none of these supplementary resources. For that student, the CBSE syllabus is not a floor — it is the ceiling. Raising that ceiling, in the ways proposed here, is an act of educational equity.

The implementation of this framework does not require CBSE to redesign its examination structure, increase contact hours, or invest in new infrastructure. It requires three things: a minor revision of the syllabus document to add AI connections and the Technology Landscape unit; an update to the project brief to include the Build-Deploy-Share model as an explicit option; and a targeted professional development programme for PGT CS and IP teachers — which, given that NCERT's DIKSHA platform already exists and is free to access, can be delivered at near-zero cost.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

One important clarification is worth making. This paper is not arguing that CBSE has ignored technology modernisation. The introduction of computational thinking and AI as a subject for Classes 3 to 8 is a meaningful step, and the shift to Python in CS and IP was a significant curriculum improvement. The argument here is specifically and narrowly about CS 083 and IP 065 at the Class XI–XII level — the two subjects taken by the students who are closest to entering the workforce or higher education, and for whom the gap between what they have learned and what the world needs is most consequential.

XI. CONCLUSION AND RECOMMENDATIONS

This paper set out with a straightforward observation: two of CBSE's most widely studied technology subjects are stronger than they appear on paper, but not yet strong enough to call themselves future-ready. Computer Science and Informatics Practices provide students with real technical skills. What they currently do not provide is the AI literacy, practical independence, global language awareness, and career visibility that would make those skills genuinely transformative. The four-pillar framework proposed here — AI integration, practical learning, language awareness, and career pathway guidance — is designed to close that distance without demanding new subjects, new examinations, or new infrastructure.

The student who completes Class XII CS or IP under this strengthened framework would leave school knowing not only how to write Python code and SQL queries, but understanding how those skills connect to Artificial Intelligence; having built and deployed at least one real-world project; knowing that the global technology landscape extends beyond the tools they have learned, and how to navigate it; and understanding, concretely, what careers their skills can lead toward. That is not a radical reimagining of the curriculum. It is the curriculum fulfilling its own stated purpose.
The following specific recommendations are offered:

- CBSE Curriculum Committee: Revise the CS 083 and IP 065 syllabus documents to include AI connection notes within existing topic descriptions, expand the Society and Ethics / Societal Impacts units to include AI ethics explicitly, and add a 4–6 period Technology Landscape unit to the Class XII syllabus of both subjects.
- CBSE Curriculum Committee: Update the Class XII project brief for both CS and IP to include the Build-Deploy-Share model as an explicitly described and encouraged option, with GitHub and Streamlit named as free, accessible tools.
- NCERT: Revise the Class XI IP Emerging Trends chapter to include a hands-on component — even a single session with Teachable Machine is sufficient to transform it from vocabulary to experience.
- NCERT / CBSE: Add a Career Pathways appendix to the Class XI and XII textbooks of both CS and IP, mapping subject skills to professional opportunities for both science and commerce stream students.
- State Education Departments and School Management: Support PGT CS and IP teachers in completing a focused AI upskilling programme — 30 hours, available free on DIKSHA and NPTEL — before the framework is introduced. Teacher preparedness is the single most important implementation factor.
- PGT CS and IP Teachers: Begin making AI connections within existing teaching, irrespective of any formal syllabus revision. The connections described in Section V of this paper can be made today, in any classroom, using the existing syllabus, with no additional tools. Framing is free.
- Researchers and Educators: Conduct classroom-based studies examining student outcomes, engagement, and career readiness under the proposed framework. Evidence is needed to support policy advocacy at the national level.

The students sitting in government senior secondary school computer labs across India are capable of far more than the current syllabus asks of them. They have access to the right foundational tools. What they need is a curriculum that connects those tools to the world they are about to enter. That is not too much to ask.

REFERENCES

[1] Ministry of Education, Government of India. (2020). National Education Policy 2020. New Delhi: Ministry of Education.
[2] Central Board of Secondary Education. (2024). Computer Science (Code 083) Curriculum, Classes XI–XII, 2024–25. New Delhi: CBSE Academic Unit.
[3] Central Board of Secondary Education. (2024). Informatics Practices (Code 065) Curriculum, Classes XI–XII, 2024–25. New Delhi: CBSE Academic Unit.
[4] Long, D., & Magerko, B. (2020). What is AI literacy? Competencies and design considerations. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (pp. 1–16). ACM.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155227

[5] Ng, D. T. K., Leung, J. K. L., Chu, S. K. W., & Qiao, M. S. (2021). Conceptualizing AI literacy: An exploratory review. Computers and Education: Artificial Intelligence, 2, 100041.
[6] Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). Envisioning AI for K–12: What should every child know about AI? Proceedings of AAAI-19, 9795–9799.
[7] Stack Overflow. (2025). Developer Survey 2025. Stack Overflow Insights. Retrieved from https://survey.stackoverflow.co/2025
[8] TIOBE Index. (2026). TIOBE Programming Community Index, January 2026. Retrieved from https://www.tiobe.com/tiobe-index/
[9] Australian Government Department of Education. (2025). China's approach to AI education in primary and secondary schools. Canberra: Department of Education.
[10] Government of Hong Kong Education Bureau. (2024). Curriculum Development and AI Education, Junior Secondary Level. Hong Kong: EDB.
[11] Singapore Ministry of Education. (2024). AI in Education: MOE's Approach. Singapore: MOE. Retrieved from https://www.moe.gov.sg
[12] UNESCO. (2022). K–12 AI curricula: A mapping of government-endorsed AI curricula. Paris: UNESCO.
[13] Central Board of Secondary Education. (2023). Annual Report 2022–2023. New Delhi: CBSE.

BIOGRAPHY

Dr Neerja Jain is a Lecturer in Computer Science at a Government Aided Senior Secondary School in Delhi, India. She holds a PhD in Computer Science along with MCA, M.Sc. (Computer Science), and B.Ed. qualifications, and has over seventeen years of classroom experience teaching both Computer Science (Code 083) and Informatics Practices (Code 065) at the senior secondary level. She has maintained a 100% board result in both subjects throughout her teaching career. Her research interests include CS curriculum development, AI in education, and the translation of technology policy into classroom practice. She has contributed to curriculum development for Delhi Model Virtual School and has served as a subject expert for Staff Selection Committees in Delhi's Zone 7.