INDIAN JOURNAL OF LEGAL REVIEW

VOLUME 6 AND ISSUE 5 OF 2026

INSTITUTE OF LEGAL EDUCATION

INDIAN JOURNAL OF LEGAL REVIEW

APIS – 3920 – 0001 | ISSN - 2583-2344

(Open Access Journal)

Journal's Home Page – https://ijlr.iledu.in/

Journal's Editorial Page - https://ijlr.iledu.in/editorial-board/

Volume 6 and Issue 5 of 2026 (Access Full Issue on - https://ijlr.iledu.in/volume-6-and-issue-5-of-2026/)

Publisher

Prasanna S,

Chairman of Institute of Legal Education

No. 08, Arul Nagar, Seera Thoppu,

Maudhanda Kurichi, Srirangam,

Tiruchirappalli – 620102

Phone : +91 73059 14348 - info@iledu.in / Chairman@iledu.in

© Institute of Legal Education

Copyright Disclaimer: All rights are reserve with Institute of Legal Education. No part of the material published on this website (Articles or Research Papers including those published in this journal) may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher. For more details refer https://ijlr.iledu.in/terms-and-condition/

INDIAN JOURNAL OF LEGAL REVIEW [IJLR – IF SCORE – 7.58]

VOLUME 6 AND ISSUE 5 OF 2026

APIS – 3920 - 0001 *(and)* ISSN - 2583-2344

Published by

Institute of Legal Education

https://iledu.in

"SOFTWARE PATENTABILITY IN INDIA: A CRITICAL ANALYSIS OF SECTION 3(K), TECHNICAL CONTRIBUTION, AND GLOBAL PATENT REGIMES IN THE ERA OF AI AND HIGH-PERFORMANCE COMPUTING"

AUTHOR - SHREYA MISHRA* & DR. SHOVA DEVI**

* STUDENT AT AMITY LAW SCHOOL, AMITY UNIVERSITY UTTAR PRADESH, LUCKNOW CAMPUS

** ASSISTANT PROFESSOR AT AMITY LAW SCHOOL, AMITY UNIVERSITY UTTAR PRADESH, LUCKNOW CAMPUS

BEST CITATION – SHREYA MISHRA & DR. SHOVA DEVI, "SOFTWARE PATENTABILITY IN INDIA: A CRITICAL ANALYSIS OF SECTION 3(K), TECHNICAL CONTRIBUTION, AND GLOBAL PATENT REGIMES IN THE ERA OF AI AND HIGH-PERFORMANCE COMPUTING", *INDIAN JOURNAL OF LEGAL REVIEW (IJLR),* 6 (5) OF 2026, PG. 350-359, APIS – 3920 – 0001 & ISSN - 2583-2344.

ABSTRACT

Software has changed so much over the years, from basic machine code back in the day to all this stuff with AI and cloud computing now, plus things like 5G that make everything faster. It is kind of wild how it has reshaped the whole world, economies and industries and even how people live. In India, this has brought up big questions about patents for software.

The Patents Act from 1970 has this Section 3(k) that says computer programs by themselves cannot be patented. It is meant to stop companies from locking up ideas that everyone should be able to use, protecting the public and keeping tech open. But the problem is, what does per se really mean? There is no clear definition, so judges and patent offices interpret it differently sometimes. That creates uncertainty, I think, for people trying to innovate or start businesses.

This makes it tough for startups especially, because they need some way to protect their ideas without jumping through too many hoops. The legislative idea behind 3(k) was cautious, to avoid monopolies, but in practice it might slow down real progress in software.

Then there are the guidelines from the Indian Patent Office on computer related inventions. They talk about needing a technical contribution, like making hardware work better or speeding up processes somehow. It is not just the code alone, but how it affects the system technically.

Cases like Ferid Allani against the Union of India helped shape this, showing that if there is a real technical effect, it might be patentable. More recent ones with Comviva Technologies build on that, evolving what counts as technical.

Comparing to other places, the US is more open with their Alice test, letting some software patents through if they are not too abstract. Europe has this technical effect idea under their convention, similar but maybe a bit stricter. India though sticks to a tighter line, which could affect how competitive we are globally, with investments and all.

That divergence matters a lot for the tech sector here. It might discourage big R and D efforts.

Now with AI and machine learning coming up strong, it gets even messier. These technologies mix algorithms with real applications, so where do you draw the line? The current rules in India do a good job preventing evergreening, like patenting small tweaks to old stuff, but they might hold back new breakthroughs too.

INDIAN JOURNAL OF LEGAL REVIEW [IJLR – IF SCORE – 7.58]

VOLUME 6 AND ISSUE 5 OF 2026

APIS – 3920 - 0001 *(and)* ISSN - 2583-2344

Published by

Institute of Legal Education

https://iledu.in

I am not totally sure, but it feels like the framework needs tweaking to keep up without losing the public interest part.

For recommendations, maybe clarify what per se covers exactly. Add a clear test for technical contribution, make those CRI guidelines stronger legally. Align a bit more with international stuff, but keep our priorities in mind.

A better system would help with certainty, I suppose, and boost innovation. It could draw more investment and help India lead in digital stuff. Though, this part is a bit messy to wrap up neatly.

KEYWOEDS: Software Patentability, Section 3(k) of the Patents Act 1970, Computer Programs *per se*, Technical Contribution, Technical Effect, Computer Related Inventions (CRI) Guidelines, Artificial Intelligence and Patent Law, Innovation and Intellectual Property Rights, Comparative Patent Regimes, TRIPS Agreement, High-Performance Computing, 5G Technology, Legal Uncertainty in Patentability, Startups and Innovation Ecosystem, Evergreening of Patents, Public Interest vs Private Rights, Digital Economy and Patent Policy

1.0 Introduction

1. Background of software and technological growth

Software development started way back, and it's come a long way since then. I mean, from basically punching cards into machines to now having AI helping write code, it's wild how much has shifted. All because of new tech popping up and people needing different things from computers. The digital stuff just keeps getting more complicated, too.

In the early days, like the 1940s and 1950s, everything was super hands-on. Programmers had to code right at the machine level, dealing with hardware directly. It was all manual, writing instructions by hand in machine language. Hardware was so limited back then that developers squeezed every bit of efficiency out of their programs. Mostly, this was for scientific calculations or military stuff, simulations and defence systems. Things like the Colossus in 1943 during the war, which was one of the first programmable computers. And then von Neumann's report in 1945 laid out the stored-program idea, which basically started modern software.

By the 1950s and into the 1960s, high-level languages changed everything. Stuff like Fortran, COBOL, LISP, and even BASIC later on made it easier to read and write code. No more just machine gibberish. Compilers and interpreters came along to turn that high-level stuff into machine code, simplifying the whole process. Grace Hopper invented the first compiler around 1952, which was huge. Usage picked up for business, like data processing and early databases. Operating systems started getting developed, too. COBOL in 1960 streamlined things a lot for business apps. IBM's System/360 in 1964 was a big shift in how software was built for mainframes.

The 1970s brought personal computers, and that opened it up to more people. Not just big companies or labs anymore. GUIs appeared, like on the Xerox Alto in 1973, making software friendlier. Then Windows and Mac OS made it usable for everyone. Home computing grew, with word processors and early games like Pong or Pac-Man. MS-DOS launched in 1980, and the Macintosh launched in 1984, both really pushing user-friendly interfaces. Dennis Ritchie created C in 1972, which influenced Unix and tons of other systems.

The Internet hit in the 1990s, turning software into this connected web thing. Tim Berners-Lee invented the World Wide Web in 1991, and that revolutionised apps. Client-server setups let users communicate across networks—browsers like Netscape, e-commerce with Amazon [549]and

[549] The Patents Act, 1970 (India), s 3(k).

INDIAN JOURNAL OF LEGAL REVIEW [IJLR – IF SCORE – 7.58]

VOLUME 6 AND ISSUE 5 OF 2026

APIS – 3920 - 0001 *(and)* ISSN - 2583-2344

Published by

Institute of Legal Education

https://iledu.in

eBay, email, all that redefined how we interact. JavaScript in 1995 became key for web stuff. Windows 95 in 1997 made GUIs standard on PCs.

Mobile took over from the 2000s on. Smartphones and tablets meant apps everywhere. App Stores from Apple in 2008 and Google made distributing software easy. Social apps like Facebook, Instagram, games like Angry Birds, and navigation with Google Maps. Even Microsoft Office went mobile. DevOps rose around 2010 for better collaboration. Docker in 2013 changed deployment with containers.

Now, cloud computing and AI are pushing things forward. Cloud-like AWS gives scalable resources for apps. AI in stuff like Siri or Alexa automates tasks. IoT for smart homes and cities. The pandemic in 2020 sped up the use of remote tools. Quantum computing is advancing, maybe for tough problems like cryptography. It seems like blockchain and DApps will grow for secure systems and finance.

Looking ahead, it's hard to say exactly, but AI might handle more of the coding and testing itself. Quantum could solve stuff we can't know, like drug discovery. Edge computing for IoT, real-time data near the source. XR for virtual worlds in games or training. Ethical stuff, privacy, green coding, that will matter more. No-code tools let non-programmers build apps. Cybersecurity has to ramp up with threats. Open source keeps collaboration going. Maybe even brain interfaces for augmenting humans. Regulations like GDPR will shape how developers work. Personalised apps predicting what you need.

All this evolution shows how software adapts to new problems. From compact military code to global AI networks. Some parts get messy when trends overlap, like mobile and cloud mixing now. I think the field will keep surprising us.

2. 5G and High-Performance Computing:

The future of software development looks pretty exciting with things like 5G getting everywhere and computing power jumping ahead so fast. It means we can run these heavy apps that need tons of data, sort of like augmented reality stuff or streaming in real time, without lags. But yeah, the field keeps shifting, so people in software have to stay flexible, always picking up new tricks and staying on top of the latest ways to do things. I think the whole point of developing software is to open up these cool tools, make them easier for regular folks to use, and somehow change how we all deal with tech and each other in the world. With tech pushing forward, it opens doors to automation, machine learning, and even ideas we havent really built yet.

Patent laws for technology have changed a lot, and that's helped inventions grow by giving rights to creators. It pushes economic stuff and science ahead because research gets shared out there. In India, these shifts show how the country went from being a colony to a big player in science and tech globally. The history starts back in British times with the first big act in 1856 for patents and designs. That set things up formally, giving inventors exclusive rights if they shared details, but it was mostly copying British rules and favoured foreign companies more than anything local.

The 1911 Act came next and replaced the old one, making it more detailed with modern bits like the British Act of 1907. It let patents go for new, useful, non-obvious products, but skipped some pharma to keep things open for people and cut down on foreign monopolies. Patents lasted 14 years, and you could oppose or revoke them, but only for stuff made in India. It ran for decades, building a base, though really it helped British interests over Indian ones.

I might be oversimplifying, but early on, India's patent system just mirrored Britain's, putting foreign companies first and not doing much for local inventors or research. The 1911 law didn't spark homegrown innovation, ignored the farming economy's needs, and had a weak setup for industries, plus limits on drug patents that kept cheap meds scarce.

Office of the Controller General of Patents, Designs and Trade Marks, *Guidelines for Examination of Computer Related Inventions (CRI)* (2017).

After independence in 1947, there was a big push to fix the old colonial setup. India wanted laws that fit its own economy and tech scene, aiming for self-reliance, building industries, and keeping public health in check with affordable drugs. No more foreign takeovers, more focus on local growth and competing worldwide.

The 1970 Patents Act was a huge turn, ditching the colonial favouritism for something that boosted Indian tech and inventions while watching out for the public. It stuck to process patents over product ones, especially in pharma, so companies could reverse engineer foreign meds and make cheap generics. That put India on the map for generic drugs, helping firms like Cipla and Ranbaxy, and even sending affordable stuff to other developing spots.

There were rules like compulsory licensing to make sure patented things got produced or imported at fair prices for everyone. Patents were capped at 14 years, and they made it tough to patent just copies, only real innovations. Then the WTO and TRIPS in 1995 hit, forcing [550]India to tweak laws for global fit. The big clash was with pharma and biotech, where India only did process patents, but TRIPS wanted product patents too.

So in 1999, they added a mailbox for filing product patents, but held off granting till 2005. That year brought major updates, allowing product patents and stretching terms to 20 years to draw in foreign cash. Still, they added section 3(d) to block minor tweaks that don't really improve things, fighting off evergreening by big drug firms and keeping drugs affordable.

The 1970 act really shielded local industries in engineering, chemicals, and pharma from foreign dominance, making sure meds stayed accessible. Before TRIPS, India kept product patents narrow for health reasons, balancing inventor rewards with public needs. Now its this ongoing pull between world rules, home development goals, and what people actually need, spilling into biotech, IT, and beyond.

On software patents, there's growing buzz worldwide since these programs drive so many new apps, but countries handle them differently. This piece looks at India's and the US systems for software, and what that spells for developers. In India, a program alone isn't patentable, but if it creates some technical effect, maybe it qualifies. Views vary on whether its obvious or has that innovative spark, and it's tricky to pin down.

The US seems more open, like in cases where software ties into real processes. Europe's got rules too, saying programs as such aren't patentable, but technical contributions might count. It feels like the debate keeps evolving, with history showing how patent shifts adapt to tech demands. Some parts get messy here, like balancing protection without stifling access.

3. Section 3(k)

Section 3(k) in the Indian Patent Act from 2002 basically excludes patents for computer programs on their own, or stuff like math models and business methods. That per se part makes it fuzzy, leading to tons of arguments in courts and with the patent office. Its why software patents are such a big deal in India right now, especially with AI popping up everywhere. AI inventions often rely on algorithms, but they can have real technical impacts, like helping machines learn from data or process things faster. Still, under this section, it might just get seen as a plain computer program and blocked.

The Indian Patent Office put out guidelines in 2016, then tweaked them in 2017 for computer-related inventions. They say if there's no real technical effect, like changing how hardware works or improving it somehow, no patent. In the US, things are different, more open, without a strict ban. Courts decide case by case, and while some old rulings said no to software patents, newer ones allow them if tied to something practical, like controlling machines

[550] Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS), 1995, arts 27–34.
Ferid Allani v Union of India (2019) SCC OnLine Del 11867.

INDIAN JOURNAL OF LEGAL REVIEW [IJLR – IF SCORE – 7.58]
VOLUME 6 AND ISSUE 5 OF 2026
APIS – 3920 - 0001 *(and)* ISSN - 2583-2344
Published by
Institute of Legal Education
https://iledu.in

in industry. The USPTO looks at whether it's beyond an abstract idea and adds something useful to the world.

India stays pretty conservative. Pure algorithms or business ideas get shut out, but software that improves tech, say in image handling or compressing data, could get through if it does more than just run on a computer. Courts have backed this, like in the Ericsson versus Intex case from the Delhi High Court, where they stressed technical effects matter most. The guidelines use a three-step check. First, is it just an algorithm or a program. Second, does it create technical effects, maybe tied to hardware or working on it? Third, does it have novelty and inventive step? The 2017 changes dropped the need for new hardware and focused on the overall contribution, which feels like a slight shift toward how places like the US do it, but India is still careful.

Inventors in India need to prove their software does something technical, not just automate basics. The US gives more room, especially for interfaces or systems in industry. A startup could patent a business app there easily, but in India, it hits roadblocks fast. Globally, software patents are changing, with India being restrictive and the US losing thanks to court decisions. AI complicates it, fitting into those exclusions or not. It seems like India might need to tweak things for AI, clarifying, per se, though I'm not sure how that plays out exactly.

The vagueness in Section 3(k) makes protecting software harder for tech firms here than in the US. Some say that's good, keeps out weak patents. Others think it slows innovation, particularly in AI, which moves quickly. It's tough to pin down a clear side.

Adding Section 3(k) was about matching TRIPS rules, so India had to cover tech patents but also shield its software industry and open source stuff. Kind of balancing outside pressure with home needs. Over time, courts have handled it differently, with mixed results and some inconsistencies. Patent offices and judges struggle with computer programs and business methods.

Take the Comviva Technologies case in 2024, about patenting token authentication for payment cards. The office said no, called it a business method or pure software. But Delhi High Court flipped it, saw technical boosts to security, and sent it back for review. Shows you can fight rejections if there's real tech innovation underneath. Then there's the Open TV case, for delivering personalised gift media digitally. Rejected by the office and court, viewed as [551]business via software with no new tech. The court did note that software can be patented if it adds a technical effect or advances tech, so outcomes vary.

For AI or software in India, inventors have to show clear technical benefits, not abstract or simple automation. Demonstrating that is key, solving a tech problem with real improvement. Like the Microsoft AI filtering interface in 2023, first rejected as routine code, but they appealed and won by explaining the technical impact. Or Niram AI's patent for breast cancer screening using thermal images and AI, less invasive than mammograms, spotting early signs. That ties to industrial use nicely.

Pure algorithm claims without tech links get rejected, as in BlackBerry's wireless admin system. For novelty, it needs big improvements, not just automating old ways. Google's instant messaging across devices got rejected at first, but the Delhi High Court in 2024 granted it for technical advances. These examples push inventors to strategise, highlighting tech value. Some call the system too tight; others say it blocks junk patents. Moving past abstract to concrete tech seems the way.

The CRI Guidelines set standards, focusing on technical effects. Courts brought in technical effect and contribution through cases like Ferid Allani and Raytheon. Technical effect means

[551] *Telefonaktiebolaget LM Ericsson v Intex Technologies (India) Ltd* (2015) 62 PTC 90 (Del).
Alice Corp Pty Ltd v CLS Bank International 573 US 208 (2014).
European Patent Convention, 1973, art 52.

practical benefits to systems or processes, like faster execution, better security, or efficient memory in software. Simple examples, improved speed in video compression, less packet loss for reliability, or quicker voice recognition with low latency.

Technical contribution is how you get that effect, like a new way to handle network packets for less delay, or an algorithm saving energy in image recognition on regular processors, or better security protocols for banking apps. In Ferid Allani versus Union of India in 2019, the Delhi HC first officially okayed the idea. Said programs are patentable if they give a tangible technical effect through operation, no need for new hardware. It was about enhancing processor performance for faster outputs in compression, applied to buffers and such. Substance over form, they said. User perks or business UI like recommendations don't count as technical, but efficiency and speed do.

Examiners use a two-step process for CRIs. Does it fix a tech problem with tech means? Does it yield a technical effect? Yes to both counters Section 3(k). When drafting, spot the tech issue, give solutions, show improvements like half the processing time, and link to hardware. Technical effect rules CRIs in India, excluding business algorithms or abstracts. Things boosting performance and security on standard hardware can be patented. It ends with comparing India, the US, and Europe for policy ideas.

The debate on software patents is big worldwide. The US and Europe have tests for tech progress, and India has stricter. Does India's way guard public good and spark innovation? In India, Section 3(k) bars computer programs per se, to stop monopolies on ideas and keep tech open.

Practically, needs technical progress. CRI Guidelines from 2017 and 2019 clarify but aren't binding, so patent processes differ. Policy-wise, India focuses on public interest, competition, open source, and cheap tech access, but that brings ambiguity and less drive for new ideas.

The US Patent Act doesn't exclude software outright, checks under Section 101 as a process or machine. Alice Corp versus CLS Bank in 2014 set the Alice Test, two steps. First, is it an abstract idea? Second, does it have an inventive concept making it eligible?

Software needs a link to specific tech execution, not just abstract. Encourages innovation without owning ideas. US aims to boost R&D, protect investments, and help tech sectors.

Europe, Article 52 EPC excludes programs as such. But EPO grants if further technical effect beyond usual software and hardware, advancing tech.

Clear line between abstract and tech inventions, predictable exams, and balances access and innovation. Europe balances, no full protection, but boundaries for real inventions.

Comparing, India has explicit exclusion in 3(k), the US has none explicit, and Europe is conditional. Test in India, technical contribution but unclear, US Alice, Europe technical effect. Predictability is low in India, moderate in the US, and high in Europe. Encourages innovation, limited here, strong US, balanced Europe. High discretion for examiners in India, judicial in the US, and standardised in Europe.

Impacts, the US and EU are more reliable, India is uncertain, scares local and foreign investment. Businesses here miss global patents, and investors go where rights are clear. Law protects access, but too restrictive hurts local development. For policy, clarify per se in 3(k), make the technical effect statutory like Europe. Adopt two step like Alice, spot abstracts, then check tech innovation. Make the CRI Guidelines binding. That shifts from tight rules, [552]protects real inventions, not abstracts. Ties to how Section 3(k) affects software innovation and startups in India.

[552] Shamnad Basheer, *Innovation, Software Patents and Indian Law* (Oxford University Press 2011).
W R Cornish, D Llewelyn and T Aplin, *Intellectual Property: Patents, Copyright, Trade Marks and Allied Rights* (Sweet & Maxwell 2019).
Lionel Bently and Brad Sherman, *Intellectual Property Law* (Oxford University Press 2014).

INDIAN JOURNAL OF LEGAL REVIEW [IJLR – IF SCORE – 7.58]

VOLUME 6 AND ISSUE 5 OF 2026

APIS – 3920 - 0001 *(and)* ISSN - 2583-2344

Published by

Institute of Legal Education

https://iledu.in

4.1 Introduction.

In today's world, the software industry stands out as a key player in driving new business ventures. Many believe that patent rights are the biggest motivators for both patent-granting agencies and innovators when it comes to making investment choices. However, Section 3(k) of the Indian Patent Act of 1970 puts a damper on software patentability in India by stating that 'no computer programme per se' can be patented. This provision is crucial for understanding how entrepreneurship and innovation will evolve within India's tech landscape.

4.2 Legislative Intent behind Section 3(k)

The exclusion introduced in Section 3(k) was designed to prevent parties from claiming exclusive rights over abstract concepts and their related algorithms. The goal was to:

- Encourage a competitive environment

- Ensure free access to software technology

- Avoid obstacles in the technology diffusion process

This clause reflects India's commitment to balancing individual interests with public benefits. However, it has also led to some practical challenges that innovation creators and business founders currently face.

4.2 Impact on Software Innovation

4.2.1 Legal Uncertainty and Innovation Risk

One of the most significant effects of Section 3(k) is the uncertainty it creates regarding the patentability of specific inventions. The phrase "computer programme per se" lacks a clear definition, resulting in varying interpretations among patent examiners and judges. As a result, patent applicants often find themselves guessing whether their software inventions will receive patent approval. This unpredictability discourages innovators from investing in long-term research and development, particularly in deep tech and new product development.

5. What is "technical contribution" under the CRI Guidelines?

5.1. Introduction

To determine if a computer-related invention is patentable in India, we need to examine its technical contribution. According to Section 3(k) of the Indian Patent Act, 1970, a patent for "computer programs per se" is prohibited. However, CRI guidelines state that a patent can be granted for computer-related inventions if they offer a technical contribution beyond the "computer programs per se." The purpose of the technical contribution requirement is to distinguish genuine technological advancements from abstract business methods, thereby striking a balance between inventors' rights and public interest. The patent applicant is expected to prove that the patent rights granted are for a software with actual, technologically grounded applications. Software-related inventions may make a technical contribution when they solve a technical problem according to CRI guidelines, meaning the claim addresses a technical issue rather than a business or administrative one. This analysis aims to shorten processing times, enhance system protection, and optimize data transfer speeds. This indicates that patent rights are best granted to software that has real practical utility or benefits. A technical effect is achieved when specific software features lead to noticeable improvements in the system's performance. These include improvements in computation speed, reduced memory usage, better hardware reliability and accuracy, enhanced data storage and retrieval efficiency, and improved process control over mechanical or industrial systems. The technical effect is typically achieved through direct interaction with the system's hardware. A software makes a claim for technical contribution if it acts as an interface between hardware components, controls sensor-based devices, manages an embedded system that operates industrial or mechanical machinery, or is an AI-based system that enhances hardware capability analysis. Any ordinary computer process

INDIAN JOURNAL OF LEGAL REVIEW [IJLR – IF SCORE – 7.58]
VOLUME 6 AND ISSUE 5 OF 2026
APIS – 3920 - 0001 *(and)* ISSN - 2583-2344
Published by
Institute of Legal Education
https://iledu.in

without the integration of novel technology will not reach the benchmark. A software claim would need an inventive method of performing the new technical feature; a new combination of technical features or inventive mechanism to achieve a technical result, i.e. it should not perform an action that can be done manually. In the CRI guideline, it is also indicated that 'mathematical method and algorithm per se, business method per se and presentation of information' are not technical contributions. Software programs that are considered "computer programs per se" are not patentable. The wording of the claims is crucial for the invention to meet the technical contribution criteria. Claims that "focus on technical architecture, on system-level improvement, on the hardware-software interface" are more likely to qualify. This prevents the patenting of abstract ideas and ensures that genuine technological innovations are protected. The patent system provides inventors with a choice between absolute exclusion and absolute patentability. However, the scope of licensing remains unclear, creating uncertainty for patent applicants. The Indian Government should revise Section 3(k) to align with TRIPS and create a more business-friendly environment. The increasing advancements in software-based inventions have highlighted the necessity for protecting computer-related inventions. The Indian Government, with a restrictive and safe approach, does not allow patentability of "computer program per se" under Section 3(k) of the Patent Act, 1970. The question of whether Section 3(k) should be amended to promote competitiveness and compliance with TRIPS standards is now a subject for universities and the global technology market, with the latter having a significant influence on the former. Under the TRIPS[553] Agreement, patentability requires an invention to be new, inventive, and industrially applicable. The agreement does not mandate patenting software inventions; instead, it allows states to define their own patentable subject matter. India's current system meets TRIPS requirements but fails to foster invention and provide a competitive business environment. 3.1 The phrase "computer programs per se" is not clearly defined in the section, which leads to varying interpretations by examiners and judges. This unpredictability negatively affects Indian startups and discourages high-technology companies from investing in the country. 5.2 Global Competitiveness Disadvantage The patentability determination of computer-related inventions in India depends upon their technical advancement. Section 3(k) of the Patent Act of 1970 states that "computer programs per se" cannot be patented, but the CRI Guidelines permit patent protection of certain software-based inventions. For patentability, the claimed invention must exhibit a technical contribution beyond mere software. The technical contribution filter works to 'exclude purely abstract algorithms, business methods'; 'allow the grant of patents to genuine technological advances' and 'maintain balance between incentives to innovate and public interest'. This shows that patent rights should only be granted to software that actually contributes technically, under practical conditions. The technical contribution is shown in a software-related invention when it solves a technical problem, as described in CRI Guidelines. The claim has a technical contribution since it solves a technical problem, instead of a business or administrative problem. The research's goal is to reduce processing time and increase system protection and data transfer effectiveness. The assessment confirms that patent rights should only be granted to software that actually contributes technically, under practical conditions. Technical effects are created when a system is enhanced technically. Recognized technical effects: quicker computation, memory reduction, more dependable or accurate hardware function, better data storage and retrieval systems, and effective control of mechanical or industrial

[553] P Narayanan, *Patent Law* (Eastern Law House 2017).
Comviva Technologies Ltd v Assistant Controller of Patents (Delhi High Court, 2024).
OpenTV Inc v Controller of Patents and Designs (Delhi High Court, 2023).

systems. The effect must be the result of hardware-software interaction, rather than just data handling. The software contributes technically via its functional role as part of specific hardware. Software has the capacity to control sensor-operated devices by performing a specific operational function. Software is used in an embedded system to control an industrial machine. AI software enhances hardware assessment capability. It does not perform an ordinary computer function. Technical contribution requires a new technical method or combination of features, an inventive process of reaching a technical outcome. A single-stage automated process that merely changes a manual task into a software version is not a technical contribution. The CRI Guidelines state clearly that 'mathematical methods and algorithms alone, business methods executed by means of computer programs, presentation of information' do not create a technical contribution. 'Programs which only fulfil the "computer programmes per se"'. The definition remains outside the patentable scope. The way that claims are drafted determines if the invention fulfils the criteria for technical contribution. Claims that "focus on the architecture, on the improvement at the system level, on the interaction between hardware and software". They are likely to achieve technical contributions more easily. This test has the role of preventing patents over abstract concepts. It allows genuine technology innovation protected by patent rights, while safeguarding the public interest. The patent system gives inventors the option of a total patent or complete patent denial. The licensing system lacks clarity due to unpredictable Patent outcomes. The Indian Government should alter Section 3(k) in accordance with TRIPS, creating a competitive business landscape. The development of software-based innovation has fostered increased discussions on patenting computer-related technology. The Indian Government's hesitation regarding the patentability of "computer programs per se" in Section 3(k) of the Patent Act 1970 is evidenced by its cautious stance. Universities have the role of deciding whether Section 3(k) should be altered to foster competitiveness within TRIPS requirements. There is increasing interdependence between the global technology markets and the TRIPS provisions. All patentable inventions must satisfy TRIPS conditions of novelty, inventive step, and industrial applicability. Countries are generally free to regulate their own patentable inventions and need not issue patents for software programs in accordance with the TRIPS agreement. They retain the right to decide on their patentable subject matters based on their own national interests. The current patent system satisfies TRIPS norms but fails to foster invention and provide a competitive business environment. 3.1 The section does not define the term 'computer program per se'. This causes unpredictable Patent outcomes as patent examiners and court judges reach diverse interpretations of Section 3(k), making India less attractive to high-technology companies.

6. The concluding remark.

Clearly, from the comparison, the present oppressive patent regime in India, which tends to secure public interest, lacks the clarity and adaptability of the American and European patent regimes. There must be a reform in an orderly fashion with a focus on best global practice, without deviating from the country's national priorities. This will boost India's innovation ecosystem considerably. Better-defined standards and rigorous testing would help India find a greater balance between technology and the public good.

Bibliography

Cornish, W. R., Llewelyn, D., & Aplin, T., Intellectual Property: Patents, Copyright, Trade Marks and Allied Rights, Sweet & Maxwell.

Bently, L., & Sherman, B., Intellectual Property Law, Oxford University Press.

WIPO, Patent Law and Practice, World Intellectual Property Organisation.

Narayanan, P., Patent Law, Eastern Law House.

Basheer, S., Innovation, Software Patents and Indian Law, Oxford University Press.

Articles and guidelines published by the Indian Patent Office.