iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81055

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

A Lightweight Architecture for Efficient Multi-User File Sharing

Mayank Giri, Divyanshu Gupta, Samiksha Singh, Himanshi Singh
DataScience (CSE) Shri Ramswaroop Memorial College of Engineering and Management Lucknow, India

Abstract: ***Nowadays more people share files online. Because of this, better ways to send data safely are needed. Email and USB drives? They struggle with big files. Central control is missing there too. Security tends to be thin. Cloud tools made saving stuff easier. Even so, who really controls your data stays unclear. Access by outsiders happens. Trusting outside companies comes up again and again.***

Out here begins a thing called Docsera - a place where people handle files together online, safely and without clutter. Built on the MERN trio, it leans on MongoDB to hold data tight, while Express and Node team up behind the scenes to manage operations smoothly. React steps in to shape how things look and move, making sure clicks and scrolls just work, no fuss. Moving around inside? Users stash files, pull them back, sort folders, send links - all from phones, laptops, anywhere with connection. Each piece fits, even when teams stretch across time zones or commute separately.

Security begins with JWT tokens, each request checked before access granted. Passwords sit safe through Bcrypt, scrambled beyond recognition. One role fits one set of powers - no more, no less - keeping data behind tight gates. Watching over it all, an admin space tracks who does what, handles stored files, checks system health without pause.

One big plus? This setup grows easily without spending too much. Because people can work together better when files move safely between them. Smooth access control keeps things tidy behind the scenes. Even with lots of users jumping in at once, it holds up fine. What sets Docsera apart comes down to three quiet strengths - tight protection, simple use, solid speed. School projects fit just right alongside office tasks or family photo swaps. No need to twist older tools into doing what they were never built for. Performance doesn't dip while keeping everything locked down properly. File handling feels lighter, faster, less tangled than before. In the end, it fills gaps others leave open by design.

Keywords: File Sharing, MERN Stack, JWT Authentication, Bcrypt, Cloud Storage, Web Application, Data Security, Multi-User System

I. INTRODUCTION

Right now, digital life moves fast because tech changes how we make, keep, and pass along data. Sharing files sits at the heart of talking and working together - whether in schools, companies, hospitals, or labs. People trade big sets of digital stuff every day: papers, photos, clips, all kinds of media. Because of that shift, strong, safe, flexible ways to share files matter more than before.

Most people still send files through email or carry them on USB sticks and portable hard drives. Yet problems pop up often with both choices. Big attachments won't go through email, versions get mixed up easily - messy when updates happen fast. Flash drives can break, disappear, or fall into the wrong hands. Working together live? Forget that with a thumb drive. No central spot means hunting down who has what, slowing everything down.

Nowadays, because cloud systems keep evolving, new ways to share files have started showing up. Services like Google Drive, Dropbox, and OneDrive let people save their work far away - on remote servers - and grab it later using any device online. Since everything links together automatically, changes appear across gadgets almost instantly. Because teamwork matters more now, tools help multiple users edit at once without confusion. Still, problems pop up even with all that convenience. Out in the digital distance, cloud setups lean heavily on outside networks - this opens doors to questions about who really guards personal data. Hacks slip through now and then, exposing private details without warning. Control? Not much of it stays with the person the data belongs to. Storage choices are made far away, behind closed tech walls.

Security gaps linger in current setups because protections often fall short. As digital attacks rise, keeping information both private and accurate turns urgent. Flimsy logins, missing encryption, or loose permissions open doors to intrusions and abuse. On top of that, oversight tools and ways to manage users usually run slow or break down. These flaws chip away at trust and stability over time.

©IJRASET: All Rights are Reserved

Getting past these issues means building a file sharing setup that is safe, easy to use, one place for everything, gives better oversight, stronger protection, smoother handling of information. When today's web tools come together, they make it possible to build platforms like this - fast, able to grow when needed. A strong base comes from using MERN Stack - MongoDB, Express.js, React.js, Node.js - shaping live, quick-reacting websites without slowing down.

Docsera isn't just another place to swap files. Built differently, it tackles weak spots in current tools by focusing on safety without slowing things down. Security kicks in right away, thanks to JWT for login checks and Bcrypt scrambling passwords behind the scenes. Uploading stuff? Done. Grabbing files later? Smooth. Share only what you mean to, setting clear limits on who sees what. Watching over actions across the system lands in the hands of admins through a dedicated control space. Each piece fits - no extra noise.

This project wants to improve how people work together online through better file sharing. With today's tools and solid protection built in, Docsera builds a system that handles rising needs across school, job, and everyday use. What stands out is its balance - simple access without losing safety. Built for real situations, it adjusts quietly whether used alone or in teams. Speed meets care where files move fast but stay guarded. Its strength lies not in features, but in how smoothly it fits into daily tasks. Behind every part is the idea: share easily, keep safe, avoid clutter.

II. LITERATURE REVIEW

Looking into earlier work means checking what has been done about file transfer setups, online storage tech, ways to keep data safe, plus tools used today to build websites. Some papers looked at how to make moving files easier, safer, faster. What comes next shows main ideas pulled from those past efforts

1) Secure File Sharing with Controlled Access

Worries about safety pop up often when files move between people. Ravi Sandhu and his team pointed out one way: giving users roles that decide what they can touch. This method cuts down on mistakes plus keeps outsiders where they belong. Another layer shows up in Bruce Schneier's take - scramble the data, whether it sits still or travels. That kind of hiding makes leaks harder even if someone grabs the wrong piece.

Security gets a boost when systems use up-to-date methods - JWT [18], for instance, plays a key role. Bcrypt [16] steps in on the password front, slowing down attacks by design. Protection of login details becomes stronger because of these layers. Cyber risks find fewer openings where such measures are applied.

2) Cloud Computing and File Sharing Systems

Out in the open, files now live on distant servers instead of desktop folders. Michael Armbrust and his team pointed out these systems grow easily when demand rises, staying online most of the time. Not far off, Mell along with Grance described the setup as a way to tap into pooled tech tools whenever needed, without long setups.

Still, research brings up issues like keeping data private and relying too much on outside platforms. Cloud setups, according to Pearson [22], tend to spark worries about who really controls information and how safe it stays. Because of these drawbacks, better protected and adjustable ways to share files become necessary.

3) Web Based App Frameworks Mern Stack

Out of today's digital setups, many rely heavily on complete toolkits like the MERN Stack. As David Flanagan points out in source nine, apps shaped with JavaScript tools tend to move fast while staying adaptable.

What Banks and Porcello found is React.js works well for building smooth, intuitive front-end designs. Behind the scenes, systems built with Node.js along with Express.js tend to grow easily as demand increases. When it comes to storing information, Chodorow points out MongoDB brings a loose, adaptable format. This kind of setup handles messy, irregular datasets without forcing rigid patterns.

Faster transfers start here when smart tools team up behind the scenes. Built strong, they handle growth without slowing down. Performance stays steady even as demands climb. Each part supports the next, quietly keeping everything moving.

4) Data Security and Cryptographic Techniques

Security around data matters because it keeps personal details safe. Writing on encryption methods, Stallings [3] points out how they guard what information stays private. Schneier [2], meanwhile, shows their role in keeping data accurate and unchanged.

Secure data moving across networks happens through Transport Layer Security (TLS) protocols [21].
On another note, tools like Bcrypt [16] keep passwords safe by turning them into scrambled forms that resist breaches.
Even with progress, plenty of setups run on shaky safeguards - proof that stacking defenses matters when swapping files. Layers help where single fixes fail.

5) software architecture system design

A well-built setup matters when making apps that can grow. As Ian Sommerville notes in source four, solid software practices help shape dependable technology.
Starting off differently each time, Pressman [5] focused on orderly methods in building software. In contrast, Fowler [8] brought forward design templates helping systems grow cleanly over time. Communication became smoother thanks to Fielding [35], who laid out RESTful principles now common across today's online platforms. Ending here feels natural.

6) Problems With Current File Sharing

Despite advancements, several challenges remain in existing systems:
Weak authentication mechanisms
Data privacy concerns in cloud platforms
Limited control over file access
Inefficient collaboration features
Dependency on internet connectivity
Working through these issues shows why better safety matters. A smoother process comes into focus when problems pile up. What looks like slow progress today might shift with smarter steps tomorrow.

7) Research Gaps Identified

Looking at past studies, some missing pieces stand out. Not every area has been covered fully. Certain questions remain unanswered still. A few parts lack clear evidence altogether. Some findings feel incomplete somehow
Lack of fully secure and centralized systems
Limited integration of authentication and file management
Poor user-friendly implementations
Lack of admin monitoring features
Inadequate scalability for large data

8) Reasons Behind Suggested Setup

The proposed system Docsera aims to overcome these limitations by:
Implementing secure authentication using JWT and Bcrypt
Providing centralized file management
Enabling controlled file sharing
Offering admin monitoring features
Ensuring scalability using MERN Stack

III. SUMMARY OF REVIEWED RESEARCH

Though strides mark how far file sharing has come, gaps remain where safety falters. Control slips through fingers like sand. Systems buckle under growing loads instead of rising to meet them. Each advance brings new hiccups no one saw coming.
A fresh approach tackles each problem through updated web tools along with strong safety layers, delivering steady performance when moving files. While built carefully behind the scenes, the result works smoothly up front.

Research Gaps and Challenges Identified

Even with better tools for sharing files and using cloud tech, today's options still fall short in key ways. Looking closer at what is already out there shows clear problems - holes in research plus hurdles yet to overcome - if anyone wants something safer, smoother, running faster, easier to use.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

1) Lack of Fully Secure Systems

Even when login features exist, stronger safeguards usually do not. Poor password rules, missing encryption, along with sloppy handling of active sessions open doors to intrusions and leaks.

2) Limited Access Control

Most file-sharing tools stick to simple rules for who can see what. Yet they tend to skip detailed controls that adjust by user role or situation. Without these features, working together safely gets tricky. Specific allowances shift poorly when needs change. Managing exactly who accesses which files remains out of reach on many systems.

3) Reliance on External Cloud Providers

Outside help runs most online file sharing platforms. Because of that, questions pop up about who really owns the data, whether it stays private, yet control often slips away since people can't fully decide how information gets handled behind the scenes.

4) Inefficient File Management Systems

Without clear structure, finding documents grows tricky over time. When folders pile up, handling them becomes a slow task. Often there is no log to show changes or where things are stored. Working through heaps of information feels messy without helpful tools. Sorting items one by one takes too long. Overloaded storage makes updates harder than needed.

5) Lack of Unified Admin Oversight

When tools lack full admin powers, oversight gets messy. Keeping track of who does what? Nearly impossible without one clear dashboard. File handling slips through cracks just as fast. System errors creep in when controls are scattered across corners. Central access would stop small issues before they spread.

6) Scalability Issues:

When too many people try to access certain file sharing setups, things start moving slower. Because they struggle with heavy loads, using them becomes tricky outside controlled settings.

7) Poor UX and UI design

Most tools feel clunky right from the start. When menus twist back on themselves or labels confuse instead of help, people slow down trying to keep up. Awkward layouts take focus away. Without clear paths forward, simple tasks drag. Getting things done becomes heavier than it should be.

8) Inadequate Use of Modern Technology

Even with tools like MERN Stack available, a lot of setups barely tap into their potential. Because of that, things run slower than needed, features stay basic.

9) Lack of Real Time Collaboration

Working together live - like changing a document at once or seeing changes right away - is something most platforms can't handle. Because of that, teams often hit roadblocks when trying to stay in sync.

10) Security versus usability balance

Security gains in numerous setups usually mean harder access for users. Still, smoother access tends to weaken protection layers. Finding middle ground - where safety does not crush ease of use - is tough work.

IV. RESEARCH GAPS IDENTIFIED

One look at today's file sharing tools shows clear flaws when it comes to handling current digital needs. Though older methods plus online storage offer easier access to files, problems still pop up around safety, user power, and smooth operation. A good number of these setups lean too much on outside providers, sparking worries about who really owns the data and whether strangers might see it. Poor login checks combined with shaky scrambling of information make leaks more likely.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Most platforms fall short when it comes to deep access controls or unified oversight. Basic permission options are all many offer, which complicates strong security enforcement. Without smart folder structures, things get messy fast.

Real-time teamwork tools often missing, causing delays. Growth becomes a problem as demands increase. Oversight suffers too - admins can't track actions clearly. Visibility slips, responsibility fades.

Right now, missing pieces in research point toward wanting something unified and locked down tight - one that handles tech issues along with user experience. What comes next? A setup called Docsera: a space where multiple users share files without friction. Security shows up early here through strong login checks, keeping things under control while organizing documents from a single hub. Structure matters too; it grows smoothly when demand rises, avoiding chokepoints later on. Web tools today make this possible, combining smart safeguards into how people exchange data digitally.

TABLE1: COMPARATIVEANALYSIS

Feature / Parameter	Traditional File Sharing (Email/USB)	Cloud-Based Systems (Google Drive, Dropbox)	FTP-Based Systems	Proposed System (Docsera)
File Sharing Method	Manual transfer (Email/USB)	Cloud-based sharing via links	Server-based transfer	Web-based centralized sharing
Security	Low (no encryption, risk of data loss)	Moderate (depends on provider)	Moderate (limited encryption)	High (JWT authentication + Bcrypt encryption)
Authentication	Not required / weak	Basic login system	Username-password based	Secure token-based authentication
Access Control	No proper control	Limited permission control	Basic access control	Role-based access control
Data Privacy	Poor	Dependent on third-party	Moderate	High (controlled environment)
Centralized Management	Not available	Available but third-party dependent	Limited	Fully centralized system
Scalability	Very low	High but paid	Moderate	High and scalable
File Size Handling	Limited	Limited (free versions)	Large file support	Efficient handling of large files
Collaboration	Not supported	Supported	Limited	Fully supported
Admin Monitoring	Not available	Limited	Limited	Full admin control and monitoring
User Interface	Basic	Good	Complex	User-friendly and responsive

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

V. CONCLUSION

Looking at how people share files shows gains made so far, yet gaps remain. Old ways like sending attachments through email or using flash drives work simply but fall short on growth potential, safety, strong performance. Handling big amounts of information becomes tough, teamwork happening live nearly impossible - both now matter a lot online.

What works once doesn't fit current needs any longer.

Getting work done from anywhere got easier because cloud tools keep files in one online spot. Take Google Drive or Dropbox - these let people save documents and pass them around using phones, tablets, or laptops. Still, handing over files to outside companies means less say over who sees what. Private details sit on servers managed by others, opening doors to leaks or spying. Even with strong features, trusting another business with your data comes with unseen downsides.

Most current setups just don't hold up when it comes to real protection. Poor login checks, shaky encoding of information, along with loose rules on who gets in - these open doors to leaks or tampering. Even when strong defenses exist, they tend to weigh down how easily people can use them. The harder something is to operate safely, the less likely it works as intended.

Most tools miss a clear way to oversee users from one spot. Without strong admin powers, watching actions or handling access gets messy. Tracking who did what becomes harder without oversight built in. Some setups struggle just to sort files well. Finding things takes too long when smart search isn't there. Large amounts of data make these flaws stand out more. Not having structure slows everything down.

Out here, fresh web tools like the MERN Stack are showing real potential against old hurdles. Not only do they support smooth app growth, but also keep operations lean and interfaces simple. Starting strong with responsive front layers, while linking smart server logic and adaptable data storage - results often land faster, feel cleaner, run quieter. The full picture? Systems work harder behind scenes so users notice less friction.

Docsera isn't just another file hub. Built after spotting gaps in existing tools, it pulls everything into one place - safe, steady, growing when needed. Security kicks in right away, thanks to JWT checks that confirm identity before entry. On top of that, passwords get scrambled with Bcrypt so they stay hidden even if breached. Access shifts depending on who you are; roles decide what you see or change. That way, control stays tight without slowing work down.

File handling gets simpler with Docsera, thanks to tools that let you move documents in and out smoothly while keeping them sorted. Oversight becomes easier when admins have a dashboard showing activity across the platform. Navigating the interface feels natural, even if someone has never used such a tool before. What stands out is how little training it takes to start using core functions.

What stands out is how the setup grows smoothly as demands rise, managing many people along with heavy data loads without slowing down. It fits well across different scenes - colleges rely on it, companies find it useful, even individuals get value. Starting from fresh tech paired with tight safeguards, better speed shows up, operations stay steady, tasks move quicker.

One thing becomes clear after looking at current tools - they work okay but fall short when it comes to mixing strong protection, ease of use, and room to grow. What stands out about the new design is how it fills those gaps without overcomplicating things. Instead of piling on extras, it focuses on what matters most today. Over time, with thoughtful updates, this setup could become something more - sharper, smarter, ready for heavier loads.

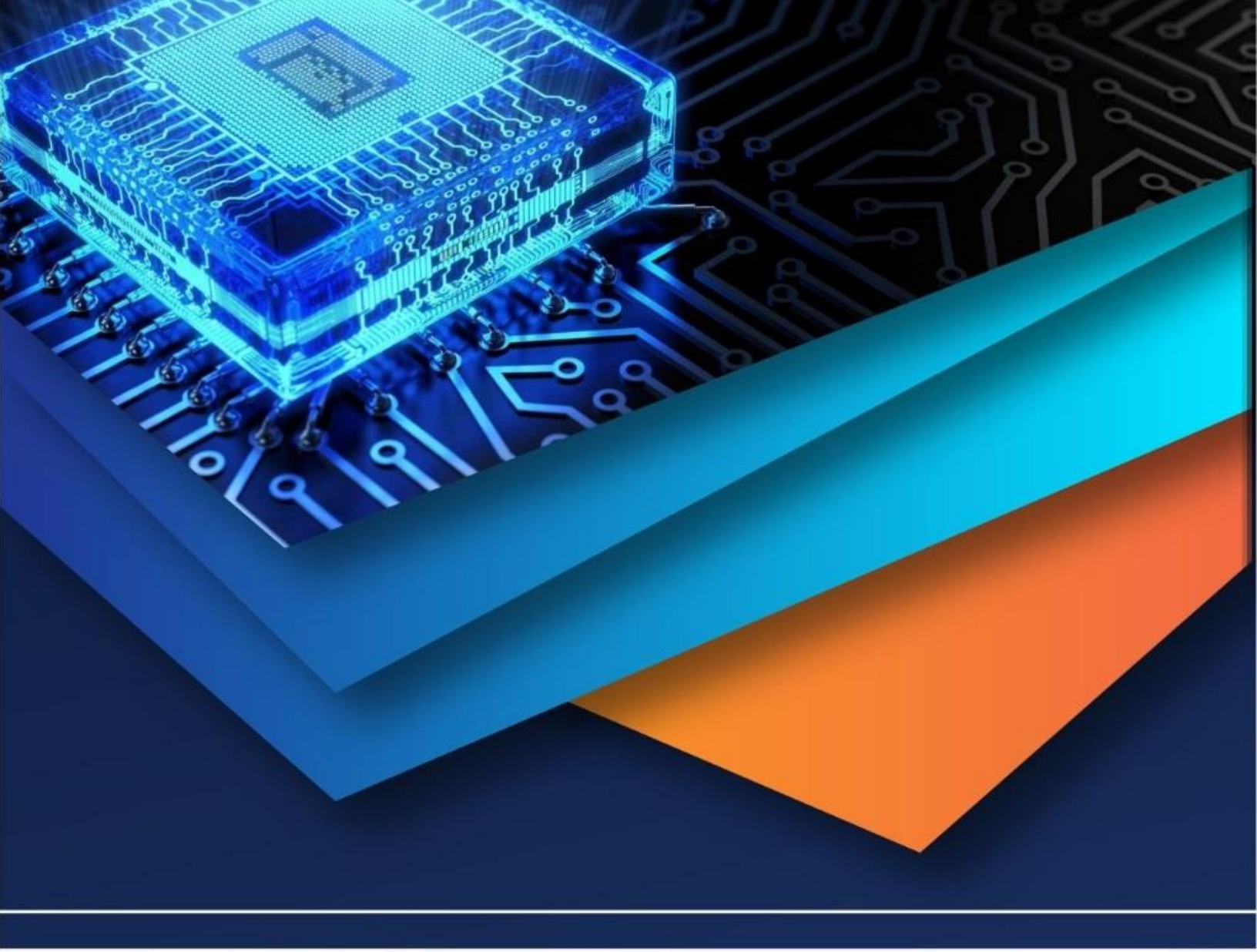
REFERENCES

[1] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.
[2] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed., New York: Wiley, 2015.
[3] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Pearson, 2017.
[4] I. Sommerville, Software Engineering, 10th ed., Pearson Education, 2016.
[5] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 8th ed., McGraw-Hill, 2020.
[6] T. Erl, Cloud Computing: Concepts, Technology &Architecture, Prentice Hall, 2013.
[7] A. Silberschatz, H. Korth, and S. Sudarshan, Database System Concepts, 7th ed., McGraw-Hill, 2019.
[8] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
[9] D. Flanagan, JavaScript: The Definitive Guide, 7th ed., O'Reilly Media, 2020.
[10] E. Brown, Web Development with Node and Express, O'Reilly Media, 2019.
[11] A. Banks and E. Porcello, Learning React, O'Reilly Media, 2020.
[12] K. Chodorow, MongoDB: The Definitive Guide, 3rd ed., O'Reilly Media, 2019.
[13] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.
[14] M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
[15] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, 2011.
[16] N. Provos and D. Mazières, "A Future-Adaptable Password Scheme," USENIX Annual Technical Conference, 1999.

©IJRASET: All Rights are Reserved

[17] D. Hardt, “The OAuth 2.0 Authorization Framework,” IETF RFC 6749, 2012.
[18] M. Jones, J. Bradley, and N. Sakimura, “JSON Web Token (JWT),” IETF RFC 7519, 2015.
[19] OWASP Foundation, “OWASP Top 10: Web Application Security Risks,” 2021.
[20] D. Eastlake, “Transport Layer Security (TLS) Protocol,” IETF RFC, 2018.
[21] S. Pearson, “Taking Account of Privacy in Cloud Computing,” IEEE CLOUD, 2009.
[22] A. Fox et al., “Above the Clouds: A Berkeley View of Cloud Computing,” 2009.
[23] E. Gamma et al., Design Patterns, Addison-Wesley, 1994.
[24] T. White, Hadoop: The Definitive Guide, 4th ed., O’Reilly Media, 2015.
[25] K. Hwang, Cloud Computing for Machine Learning and Cognitive Applications, MIT Press, 2017.
[26] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[27] A. Tanenbaum and D. Wetherall, Computer Networks, 5th ed., Pearson, 2011.
[28] J. Kurose and K. Ross, Computer Networking: A Top-Down Approach, 7th ed., Pearson, 2017.
[29] R. Fielding, “Architectural Styles and the Design of Network-Based Software Architectures,” PhD Thesis, 2000.
[30] S. Newman, Building Microservices, O’Reilly Media, 2015.
[31] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, Addison-Wesley, 2012.
[32] D. Merkel, “Docker: Lightweight Linux Containers,” Linux Journal, 2014.
[33] B. Burns, J. Beda, and K. Hightower, Kubernetes: Up and Running, O’Reilly Media, 2019.
[34] Amazon Web Services, “AWS Cloud Architecture,” White Paper, 2020.
[35] Google Cloud, “Cloud Storage Overview,” Google Documentation, 2021.
[36] Microsoft Azure, “Azure Storage Documentation,” Microsoft Docs, 2021.
[37] IBM, “Cloud Computing Security Framework,” IBM Research, 2020.
[38] Cisco, “Data Security and Privacy in Cloud Computing,” Cisco White Paper, 2019.
[39] ISO/IEC 27001, “Information Security Management Systems,” ISO, 2013.
[40] NIST, “Cybersecurity Framework,” 2018.
[41] GitHub, “Version Control System Documentation,” 2022.
[42] Dropbox Inc., “File Sharing System Documentation,” 2020.
[43] Google, “Google Drive Technical Overview,” 2021.
[44] Box Inc., “Enterprise File Sharing White Paper,” 2021.
[45] DigitalOcean, “Scalable Web Applications Guide,” 2022.
[46] Oracle, “Cloud Infrastructure Documentation,” 2021.
[47] SAP, “Enterprise Data Management Systems,” 2020.
[48] Red Hat, “Open Source Cloud Technologies,” 2021.
[49] Coursera, “Full Stack Web Development Specialization,” 2022.
[50] Udemy, “Complete Web Development Bootcamp,” 2021.
[51] M. Satyanarayanan, “Cloud Offloading,” ACM SIGOPS, 2015.
[52] A. Vaswani et al., “Attention is All You Need,” NIPS, 2017.
[53] Facebook Engineering, “Scaling React Applications,” 2020.
[54] Node.js Foundation, “Event-Driven Architecture,” 2021.
[55] MongoDB Atlas, “Cloud Database Services,” 2022.
[56] Express.js Documentation, “Routing and Middleware,” 2021.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)