

## A Quantitative and Qualitative Analysis of Popular Integrated Development Environments for Frontend Development

Patrick James Raj

Dublin, California, USA

### ABSTRACT

This paper presents a comprehensive analysis of popular integrated development environments (IDEs) used in front-end development. Focusing on both market usage data and qualitative comparisons of key feature sets, we examine Visual Studio Code, WebStorm, Sublime Text, IntelliJ IDEA, Zed, Eclipse Theia, and Visual Studio Community edition. We discuss adoption trends, core strengths, and usability aspects—including ease of configuring extensions and plugins—while also reporting approximated performance and startup measurements seen on our test system. This paper offers practical guidance for developers and organizations in selecting an IDE that best fits their workflow and project needs.

### \*Corresponding author

Patrick James Raj, Dublin, California, USA.

**Received:** June 09, 2025; **Accepted:** June 14, 2025; **Published:** June 25, 2025

**Keywords:** Integrated Development Environment, Frontend Development, Market Share, Usability, Performance, Plugin Ecosystem, Cross-Platform

### Introduction

In recent years, the software development landscape has undergone a significant transformation. There was once a clear distinction between streamlined, lightweight code editors and robust, feature-heavy environments such as Visual Studio. Today, however, the rapid adoption of tools such as Visual Studio Code and Sublime Text has blurred these boundaries. Many traditionally minimalist editors have evolved to include advanced capabilities, such as integrated source control, intelligent code suggestions, live collaboration, and real-time debugging, that were once exclusive to full-scale IDEs.

In this paper, we refer to all the selected tools as IDEs, reflecting the modern reality in which developers expect a rich set of features regardless of historical classification. Notably, our comparison includes two prominent offerings from JetBrains—WebStorm and IntelliJ IDEA. Although both are built on a similar plugin architecture, they cater to unique needs: WebStorm is purpose-built for modern web development (supporting JavaScript, TypeScript, and popular frontend frameworks), while IntelliJ IDEA serves as an all-purpose IDE with robust support for JVM-based and multi-language projects. Additionally, we include Visual Studio Community edition—a free, full-featured offering from Microsoft. However, unlike the other included tools, Visual Studio Community edition is available only on Windows (Visual Studio for Mac is no longer supported as of June 2024 and there is no Linux version); Mac users must therefore use Visual Studio Code or run the Windows version via a virtual machine.

To further aid developers, we analyse how each tool supports focused front-end development tasks such as rapid prototyping,

real-time editing, and integrated debugging.

### Strengths, Key Features and Cross-Platform Install Details

This section outlines each IDE's distinct strengths and key features for frontend development, along with specific details on installer download sizes, installed footprints, and any notable platform-specific requirements. (All the tools, except Visual Studio Community edition, are fully cross-platform.)

#### Visual Studio Code (VS Code)

1. **Popularity & Adoption:** Widely used, driven by its active open-source community and free availability.
2. **Core Strengths:** A vast extension ecosystem supporting every modern front-end technology; integrated Git support and advanced AI-assisted features (e.g., GitHub Copilot).
3. **Key Features for Frontend Development:** Intuitive interface with real-time collaboration (e.g., Live Share) and comprehensive language support for JavaScript, TypeScript, HTML, and CSS.
4. **Install Size & Platform Requirements**
  - **Download Size:** Typically, under 200 MB on Windows, Linux, and macOS (103 MB on Windows as of the date of publication).
  - **Installed Footprint:** Less than 500 MB across all platforms.
  - **Requirements:** Standard current OS versions; no special prerequisites.

#### WebStorm

1. **Popularity & Usage:** Favoured in professional environments with comprehensive web development needs.
2. **Core Strengths:** Deep integration with frameworks such as React, Angular, and Vue and robust debugging, testing, and code-refactoring tools.
3. **Key Features for Frontend Development:** Specialized

support for JavaScript and TypeScript with built-in integration for build tools and version control.

4. **Licensing & Pricing:** Subscription-based at approximately \$29.90 per month (discounted annual options around \$249 per year). Perpetual fallback license available after a defined continuous subscription period; no free community edition.
5. **Install Size & Platform Requirements**
  - **Download Size:** ~800 MB on Windows; ~1 GB on macOS/Linux.
  - **Installed Footprint:** ~3.5 GB on macOS/Linux; ~5 GB recommended on Windows.
  - **Requirements:** Modern multi-core CPU and ample disk space.

#### Sublime Text

1. **Popularity & Market Position:** Renowned for its exceptional speed and minimalist design.
2. **Core Strengths:** Extremely fast startup and low resource consumption; highly customizable via a robust plugin ecosystem managed with Package Control.
3. **Key Features for Frontend Development:** Distraction-free interface ideal for quick edits in HTML, CSS, and JavaScript with direct configurability via JSON-formatted settings.
4. **Install Size & Platform Requirements**
  - **Download Size:** Under 50 MB on all platforms.

- **Installed Footprint:** Under 100 MB.
- **Requirements:** Minimal system specifications; optimized for modest hardware.

#### IntelliJ IDEA

1. **Popularity & Adoption:** Dominates within the Java ecosystem and is increasingly used for multi-language projects, including frontend development.
2. **Core Strengths:** Offers advanced code insights, smart refactoring, and comprehensive debugging with deep integration across a wide range of development tools.
3. **Key Features for Frontend Development:** Strong support for web technologies (HTML, CSS, JavaScript, and TypeScript) along with productivity features like local history and intelligent inspections.
4. **Licensing & Pricing**
  - **Ultimate Edition:** Subscription-based (~\$19.90/month with annual discounts) plus perpetual fallback license.
  - **Community Edition:** Free, open-source, suitable for basic development (lacks some Ultimate features).
5. **Install Size & Platform Requirements**
  - **Download Size:** ~2–3 GB.
  - **Installed Footprint:** ~2–3 GB on Windows, Linux, and macOS.
  - **Requirements:** Robust system with sufficient storage.

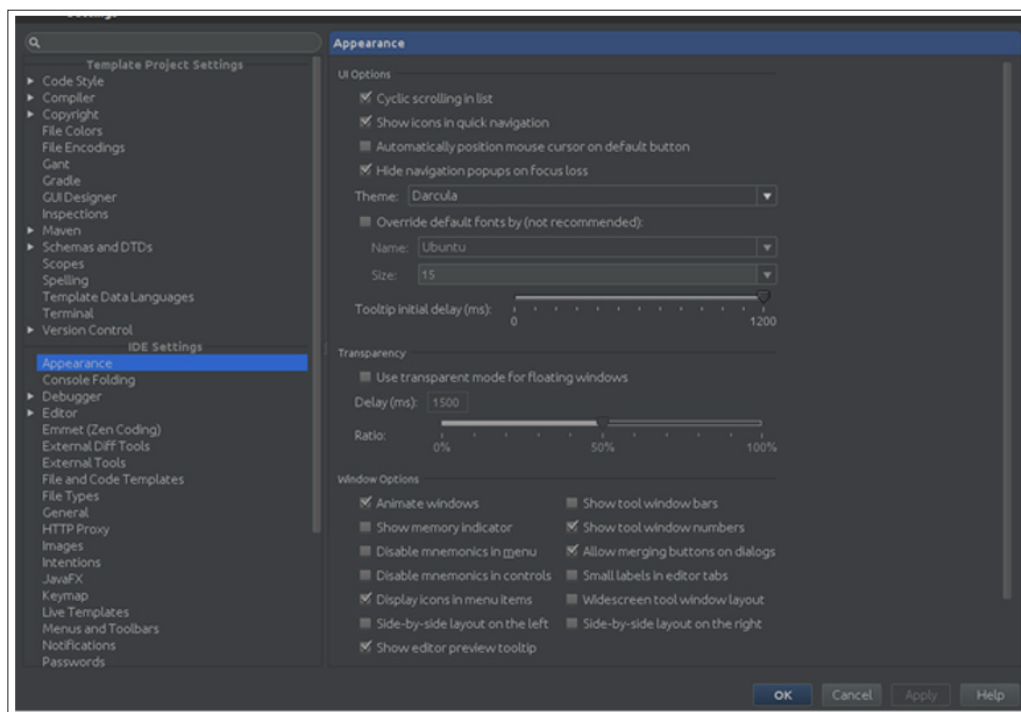


Figure 1: IntelliJ IDEA Appearance Settings in Ubuntu 20.04  
Source: askubuntu.com

#### Zed

1. **Popularity & Emerging Status:** Gaining recognition as a modern, distraction-free IDE.
2. **Core Strengths:** Minimalistic interface engineered for rapid startup and efficient coding; innovative extension framework aligned with modern web-based development trends.
3. **Key Features for Frontend Development:** Designed for quick launches with a streamlined, uncluttered workspace ideal for UI development.
4. **Install Size & Platform Requirements**
  - **Download Size:** Under 100 MB on all platforms.
  - **Installed Footprint:** Minimal, ensuring fast installation and updates.
  - **Requirements:** Standard system requirements without unusual dependencies.

Eclipse Theia

- 1. **Popularity & Niche Adoption:** Suited for cloud-based and custom IDE deployments; gaining traction for its collaborative and remote workflow capabilities.
- 2. **Core Strengths:** Flexible, modular, open-source architecture allowing extensive customization; compatibility with VS Code extensions expands its functionality significantly.
- 3. **Key Features for Frontend Development:** Perfect for remote and collaborative development with versatile configuration options via both graphical interfaces and JSON-based settings.
- 4. **Install Size & Platform Requirements**
  - **Download Size:** Windows ~320 MB; Linux ~375 MB; macOS ~400 MB.
  - **Installed Footprint:** Windows ~250–300 MB; Linux comparable; macOS ~700 MB.
  - **Requirements:** Standard Electron-based application prerequisites.

Visual Studio Community Edition

- 1. **Popularity & Adoption:** A free, full-featured IDE from Microsoft, widely used by individuals and small teams within the Microsoft ecosystem.
- 2. **Core Strengths:** Robust debugging, IntelliSense, integrated version control, and seamless Azure integration.
- 3. **Key Features for Frontend Development:** Full support for modern web applications (HTML, CSS, JavaScript, TypeScript), advanced project templates, code analysis, and testing frameworks.
- 4. **Licensing & Pricing:** Free for individual developers, open-source projects, academic research, and small teams; perpetual license with no subscription.
- 5. **Install Size & Platform Requirements**
  - **Download Size (Windows):** ~1.5–2 GB, depending on workloads.
  - **Installed Footprint (Windows):** Minimal install ~15–20 GB disk space.
  - **Platform Availability:** Windows only (no macOS/Linux).
  - **Requirements:** Windows 10 or later.

Data Collection, Popularity and Market Usage Metrics

While technical performance metrics provide one perspective, this paper focuses on qualitative aspects such as market adoption, feature strengths, and overall usability. Data on market share and usage were aggregated from reputable sources including developer surveys and industry reports.

Recent surveys show that over 50% of developers use Visual Studio Code as their primary IDE [1]. WebStorm is immensely popular among professional web teams; Sublime Text keeps a niche for its exceptional speed and minimalism; and IntelliJ IDEA commands a large share among Java and multi-language developers. Additionally, emerging tools such as Zed and platforms such as Eclipse Theia continue to carve out their niches. Visual Studio Community edition, though limited to Windows, is estimated to hold an approximate market share of around 6–7% among developers working within the Microsoft ecosystem.

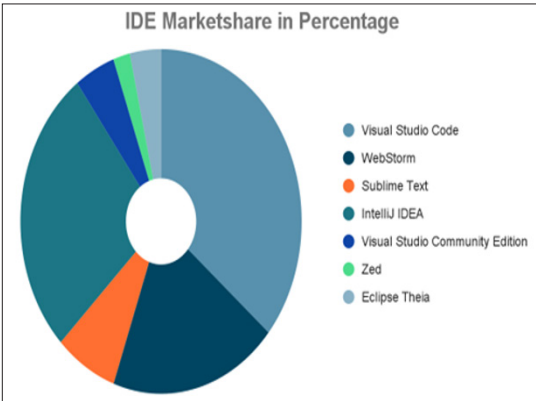


Figure 2: IDE Marketshare

Table 1: Aggregated Market Share and Usage Metrics for Selected IDEs

IDE	Usage/Market Share (%)	Data Source / Reference
Visual Studio Code	50.5%	[1]
WebStorm	25–30%	[2]
Sublime Text	10%	[1] + other reports
IntelliJ IDEA	35–40%*	[3]
Visual Studio Community	6–7% (est.)	Estimated
Zed	2–3% (emerging)	Early reports
Eclipse Theia	5%	[4]

Performance Metrics

To assess the responsiveness and resource consumption of each IDE, systematic performance tests were conducted on a system running Windows 10. The measurements below, captured under near-ideal conditions on a ThinkPad P51, serve as a baseline for understanding startup speed and memory usage.

Test System Configuration

- **Processor:** Quad-Core Intel® Core™ i7 (8th Generation)
- **Memory:** 64 GB DDR4 RAM
- **Storage:** NVMe SSD
- **Graphics:** NVIDIA Quadro M1200 (4 GB VRAM)
- **Operating System:** Windows 10

Table 2: Measured Performance and Startup Figures

IDE	Startup Time (sec)	Memory Usage (MB)
VS Code	3	200
WebStorm	6	500
Sublime Text	1.5	100
IntelliJ IDEA	5	400
Zed	2	150
Eclipse Theia	4	250

### Additional Observations

- **VS Code:** Moderate resource usage with stable startup.
- **WebStorm:** Higher startup overhead, reflecting extensive feature set.
- **Sublime Text:** Extremely fast and lightweight—ideal for quick edits.
- **IntelliJ IDEA:** Robust feature set with moderate latency.
- **Zed:** Quick startup and efficient resource management.
- **Eclipse Theia:** Reasonable startup times; well-suited for remote configurations.

### Extensibility and Community Support

Extending an IDE with plugins and tapping into its community is often what turns a good editor into a perfect fit for your workflow. For front-end developers who juggle PHP back-ends alongside JavaScript frameworks, it is critical that extensions install cleanly, play well together, and do not bog down your day-to-day coding.

#### Extensibility

##### • Visual Studio Code

With its built-in Marketplace, VS Code makes adding Intelephense for PHP or ESLint/Prettier for JavaScript a matter of one or two clicks. TypeScript and JavaScript tooling come pre-installed, and even if you load twenty or more extensions—framework snippets, debuggers, linters—memory use hovers around 250 MB and cold startups stay under five seconds. Workspace-specific enable/disable means side projects do not inherit plugins they do not need [5].

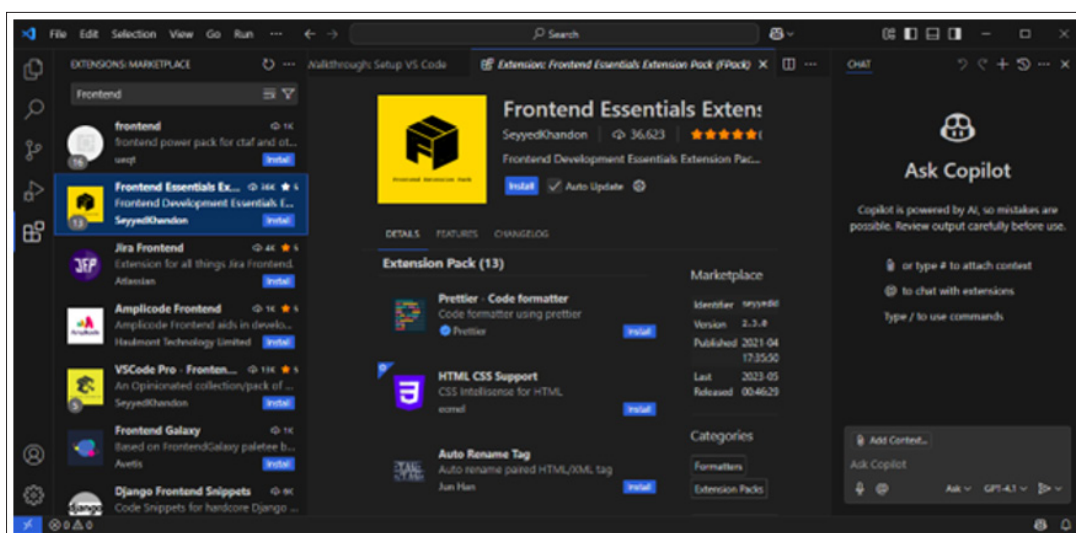


Figure 3: VSCode offers Copilot AI Integration and a comprehensive set of Extensions

##### • WebStorm & IntelliJ IDEA

Most web-dev plugins live in Settings out of the box: React, Vue, Angular, Node.js debugging—all ready to go. If you need PHP, you drop in an extra PhpStorm plugin or switch to the full PhpStorm IDE. It integrates seamlessly, but each extra language pack nudges startup from about six seconds toward eight [6].

##### • Sublime Text

Package Control is straightforward—either edit its JSON file or summon the Command Palette to fetch PHP Companion, JavaScript Enhancements, LSP bridges and more. Startup stays under two seconds and RAM under 120 MB, but setting up live linting or advanced refactoring can mean chaining together several packages by hand [7].

##### • Zed

A lean newcomer, Zed's registry covers basic syntax highlighting and formatting. You will find a few PHP and JS packages, but nothing like a full debugger. Even with a dozen plugins, RAM use stays below 160 MB and launch hovers near two seconds. The trade-off is an ecosystem that is just getting started [8].

##### • Eclipse Theia

Since Theia can consume VS Code extensions, you simply drop VSIX files into its plugins folder or install via its own manager. Intelephense, Prettier, Xdebug adapters all behave as they do in VS Code. Load twenty extensions and expect memory around 270 MB and startup near five seconds. Occasionally you will pin versions to resolve conflicts, but you gain cloud-ready, container-friendly workflows [9].

##### • Visual Studio Community Edition

First-party JavaScript tooling (Node.js, NPM integration, ESLint) comes via the VSIX gallery, and PHP support installs through third-party packages like DevSense's PHP Tools. Those extensions run an external installer, yet once in place you benefit from native IntelliSense and Xdebug-style debugging. Heavy workloads—database managers, live testing—can double memory and add a few seconds to startup, but Microsoft's installer takes care of dependencies behind the scenes [10].



**Community Support**

A thriving community is not just nice to have—it is how your IDE keeps pace with ever-shifting web standards. For most of these IDEs Stack Overflow provides a readily usable support system as well as a rich archive of existing issues and related solutions. One just needs to look for the respective tags [11]

- **Visual Studio Code**  
Boasts the largest ecosystem: a GitHub repo with thousands of contributors, weekly releases, and a Marketplace full of peer-reviewed extensions. Tens of thousands of “vscode” questions on Stack Overflow, comprehensive Microsoft Learn tutorials, plus active Discord and Gitter channels make it trivial to find answers or request features [12].
- **WebStorm & IntelliJ IDEA**  
JetBrains backs both with in-depth docs, an official YouTrack tracker, webinars, and a dedicated forum. The “webstorm” and “intellij-idea” tags on Stack Overflow see steady traffic, and plugin authors often collaborate directly with JetBrains engineers, ensuring swift bug fixes and compatibility updates [13].
- **Sublime Text**  
Its community gathers on an official forum and subreddit. Although Sublime HQ staff rarely post public updates, Package Control exposes download counts so you can judge plugin health. Most issue triage happens on GitHub, driven by a handful of passionate experts [14].
- **Zed**  
Still in early alpha, Zed’s main gathering spot is its Discord server and GitHub issues page. Turnaround on bug reports is fast, but outside official channels you will find only a smattering of blog posts—perfect if you enjoy contributing to nascent projects, less so if you need polished tutorials [15].
- **Eclipse Theia**  
Leans on the Eclipse Foundation’s mailing lists, bi-annual conferences, and “eclipse-theia” questions on Stack Overflow. Many VS Code extension developers cross-post patches, so one can take help from the broader VS Code community’s expertise [16].
- **Visual Studio Community Edition**  
Taps Microsoft’s decades-old support network: Microsoft Docs, MSDN blogs, the Developer Community portal, and over 200,000 “visual-studio” questions on Stack Overflow. Certified VSIXs in the Marketplace come with user ratings, ensuring you pick stable, well-supported extensions [17].

**Rating Summary**

To derive these ratings, we assessed each IDE’s plugin ecosystem (diversity, installation/update ease, performance impact, compatibility) and community vitality (documentation completeness, forum and issue-tracker activity, response speed). Each dimension was mapped onto a 1–5 scale—1 indicates limited support, 5 indicates extensive, well-maintained support.

IDE	Extensibility (1–5)	Community Support (1–5)
Visual Studio Code	5	5
WebStorm	4	4
Sublime Text	3	3
IntelliJ IDEA	4	5
Zed	2	2
Eclipse Theia	4	3
Visual Studio Community	3	5

**Discussion and Conclusion**

This Analysis confirms that the open-source model of Visual Studio Code and its expansive extension ecosystem have enabled it to secure a dominant market position among frontend developers. WebStorm delivers a robust, integrated toolset ideal for professional teams, while Sublime Text is highly regarded for its exceptional speed and minimalist design. IntelliJ IDEA excels in both Java and multi-language projects—further bolstered by the availability of a free Community Edition—and Visual Studio Community edition serves as a strong, free full-featured option within the Microsoft ecosystem (although it is only available on Windows, since Visual Studio for Mac is no longer supported as of June 2024 and no Linux version exists). Additionally, emerging tools such as Zed, with its minimalist design, and Eclipse Theia, with its collaborative, cloud-centric capabilities, illustrate evolving trends in IDE design.

Except for JetBrains’ WebStorm, all the other tools can be used without any licensing issues. WebStorm does offer a 30-day trial along with a perpetual free edition for personal usage. The free usage explains the large usage numbers for both VS Code and JetBrains’ own IntelliJ IDEA. Ultimately, the choice of IDE depends on both personal and organizational needs, whether for rapid prototyping or full-scale enterprise development. Future research may benefit from more granular user surveys and longitudinal studies to further elucidate emerging trends, particularly as cross-platform performance and AI-assisted coding continue to shape the development landscape.

**Acknowledgments**

The authors want to thank the development communities and data providers for their invaluable insights, including:

- The Stack Overflow Developer Survey 2021, which provided key usage statistics [1].
- The JetBrains Developer Ecosystem Survey 2021, detailing the usage of IDEs among professional developers [2].
- The Java IDEs in 2024 Survey on Baeldung, which highlighted IntelliJ IDEA’s strong presence in the Java community [3].
- Market share insights from Datanyze [4].
- Qualitative analyses from industry blogs such as Frontend Dogma and the JetBrains Blog: IDEs vs. Code Editors [18,19].

**References**

1. (2021) Stack Overflow Developer Survey. Stack Overflow <https://insights.stackoverflow.com/survey/2021>.
2. (2021) JetBrains Developer Ecosystem Survey. JetBrains <https://www.jetbrains.com/lp/devecosystem-2021/>.
3. Parashiv E (2024) Java IDEs in 2024 — Survey on Baeldung. Baeldung <https://www.baeldung.com/java-ides-in-2024>.
4. (2025) Datanyze Integrated Development Environments Market Share. Datanyze <https://www.datanyze.com/market-share/ide--444>.
5. (2025) Visual Studio Code Marketplace. Microsoft <https://marketplace.visualstudio.com/vscode>.
6. (2025) JetBrains Plugins Repository. JetBrains <https://plugins.jetbrains.com/>.
7. (2025) Package Control — Sublime Text. Package Control <https://packagecontrol.io/>.
8. (2025) Zed Extension Registry. Zed <https://zed.dev/extensions/>.
9. (2025) Authoring Extensions — Eclipse Theia. Eclipse Foundation [https://theia-ide.org/docs/authoring\\_extensions/](https://theia-ide.org/docs/authoring_extensions/).
10. (2025) Visual Studio Marketplace. Microsoft <https://marketplace.visualstudio.com/vs>.
11. (2025) Stack Overflow Tag Info. Stack Overflow <https://>

- stackoverflow.com/questions/tagged/.
12. (2025) Visual Studio Code – Documentation. Visual Studio <https://code.visualstudio.com/docs>.
13. (2025) JetBrains Support. JetBrains <https://www.jetbrains.com/help/>.
14. (2025) Support. Sublime Text <https://www.sublimetext.com/support>.
15. (2025) Zed Documentation. Zed <https://zed.dev/docs/>.
16. (2025) Eclipse Theia IDE Support. Theia Github <https://theia-ide.org/support/>.
17. (2025) Visual Studio Support. Microsoft <https://visualstudio.microsoft.com/support/>.
18. (2025) Frontend Dogma — The 4 Top IDEs for Developers in 2025. Frontend Dogma <https://frontenddogma.com/posts/2025/the-4-top-ides-for-developers-in-2025/>.
19. Watson D (2024) JetBrains Blog: IDEs vs. Code Editors: Everything You Need to Know. JetBrains <https://blog.jetbrains.com/webstorm/2024/03/ides-vs-code-editors/>.

**Copyright:** ©2025 Patrick James Raj. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.