



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 Issue: II Month of publication: February 2022

DOI: <https://doi.org/10.22214/ijraset.2022.40247>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

MERN Stack: Technologies Used for Web Development

Akshata Vhandal¹, Sanyam Gandhak², Saundarya Karhale³, Sandipkumar Prasad⁴, Prof. Sudesh A. Bachwani⁵

^{1, 2, 3, 4, 5}Department of Computer Engineering

^{1, 2, 3, 4}Government College of Engineering Yavatmal, Maharashtra, India,

⁵Assistant Professor, Government College of Engineering Yavatmal, Maharashtra, India

^{1, 2, 3, 4}Dr. Babasaheb Ambedkar Technological University Lonere, India

I. INTRODUCTION

Web development comes with a huge set of rules and techniques every website developer should know about. If you want a website to look and function as you wish them to, you need to get familiar with web technologies that will help you achieve your goal. Developing an app or a website typically comes down to knowing 3 main languages: JavaScript, CSS, and HTML. And while it sounds quite complicated, once you know what you are doing, understanding web technology and the way it works becomes significantly easier. Fret not if it is not coming easily to you immediately. You may need more time, training, and patience to dive deeper into the subject, but you'll end up with a good understanding eventually. We present you with an introduction to web technologies and the latest web technologies list hoping it will make things at least a bit easier for you. Now, let's take a look.

A. What is the MERN Stack?

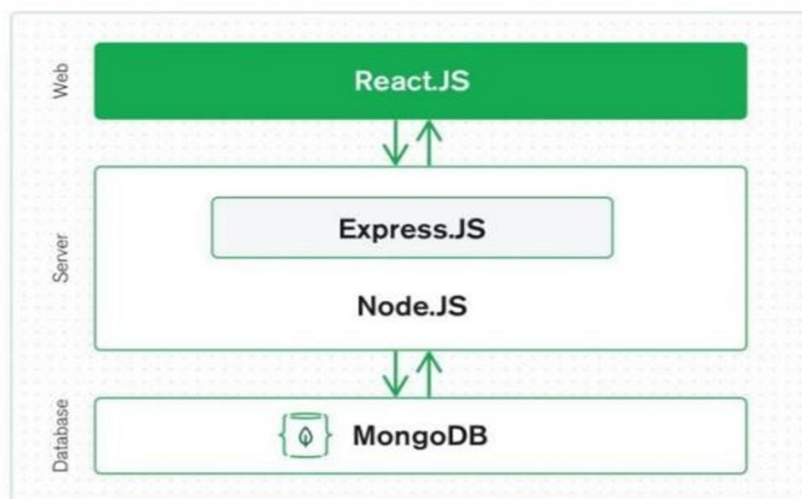
MERN stands for MongoDB, Express, React, Node, after the four key technologies that make up the stack.

- 1) MongoDB - document database
- 2) Express(.js) - Node.js web framework
- 3) React(.js) - a client-side JavaScript framework
- 4) Node(.js) - the premier JavaScript web server

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js the popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

B. How does MERN Stack Work?

The MERN architecture allows you to easily construct a 3-tier architecture (frontend, backend, database) entirely using JavaScript and JSON.



C. *React.js Front End*

The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML.

React's strong suit is handling stateful, data-driven interfaces with minimal code and minimal pain, and it has all the bells and whistles you'd expect from a modern web framework: great support for forms, error handling, events, lists, and more.

D. *Express.js and Node.js Server Tier*

The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a "fast, unopinionated, minimalist web framework for Node.js," and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

By making XML HTTP Requests (XHRs) or GETs or POSTs from your React.js front-end, you can connect to Express.js functions that power your application. Those functions in turn use MongoDB's Node.js drivers, either via callbacks for using Promises, to access and update data in your MongoDB database.

E. *MongoDB Database Tier*

If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you're going to want a database that's just as easy to work with as React, Express, and Node.

That's where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval. Again, if you're building in the cloud, you'll want to look at Atlas. If you're looking to set up your own MERN stack, read on!

II. MERN STACK COMPONENTS

A. *MongoDB*

MongoDB is a NoSQL database where each record is a document comprising of key-value pairs that are similar to JSON (JavaScript Object Notation) objects. MongoDB is flexible and allows its users to create schema, databases, tables, etc. Credentials that are special by a primary key make up the basic unit of MongoDB. Once MongoDB is installed, users can make use of Mongo shell as well. Mongo shell incorporates a JavaScript interface through which the users can interact and carry out operations like querying, updating records, deleting records.

B. *Express.js*

Express is a back-end web application framework for the Node.js framework. Rather than writing the code using Node.js and creating loads of Node modules, Express makes it simpler and easier to write the back-end code. Express helps in designing great web applications and APIs.

Express supports many middleware's which makes the code shorter and easier to write. As an alternative to writing full webserver code by hand on Node.js directly, developers use Express to simplify the task of writing server code. There's no need to repeat the same code over and over. The Express framework is considered for building robust web applications and APIs. It's known for its fast speed and modest structure, with many features available as plugins.

C. *ReactJS*

React is a JavaScript library that is used for building user interfaces. React is used for the development of single-page web applications, complex software applications, and mobile applications. React can to switch rapidly changing data. React allows the client to code in JavaScript and create UI components. React is an open-source JavaScript library that can be used for creating views rendered in HTML. Unlike AngularJS, React is not a framework. It is a library.

D. *Node.js*

A cross-platform JavaScript runtime environment which was initially built for Google Chrome and later open-sourced by Google in 2008. It's intended to build scalable network applications and can execute JavaScript code outside of a browser. Node.js provides a JavaScript Environment which allows the user to run their code on the server (outside the browser). Node pack manager i.e. npm allows the user to choose from thousands of free packages (node modules) to download.

III. COMBINING TECHNOLOGIES AND INVESTIGATING HOW THEY INTERACT



We looked at all four technologies that make up the MERN stack development in the previous section. While each of these technologies is wonderful to work with on its own, integrating them into a web application that runs like clockwork is fantastic. This is what we'll be learning about in this part. Knowing how to use each technology is half the battle; the other half is piecing it all together into something useful. What could be better than a visual depiction of a MERN stack development web application in which the user interacts with the frontend, which then interacts with the backend and database?

Let's look at an example to help us comprehend the entire procedure. Assume we've created a fantastic e-commerce website that sells clothing. Our website is currently being used by a consumer who is looking for some sneakers. On the landing page, there is a link that sends the user to the shoes page. So, what's the best way to retrieve data from the backend? Let's take things slowly at first.

- 1) The user is on our website's landing page, which was developed with React.
- 2) The user selects the link to go shopping for sneakers. Because we have a single page application, we render the shoes page without refreshing the page.
- 3) At this time, we don't have any shoe data, therefore the state is empty. To retrieve the data, we issue an API request to our backend.
- 4) Because the process of retrieving data from our database is asynchronous, which means it may take some time to complete, we show the user a loading GIF while the data for shoes is being obtained.
- 5) In the backend, ExpressJS examines the endpoint (route) we've reached and calls the relevant controller function to obtain the shoes' data.
- 6) We utilise mongoose within this controller method to query our database, retrieve the data, and return it in JSON format (JavaScript Object Notation).
- 7) This JSON data is delivered back to our React frontend, where we can use it to update the state with the freshly acquired information.
- 8) React will re-render the components that rely on our state now that it has been changed, so we replace our loading GIF with the shoes information.

IV. WHAT IS A WEB DEVELOPMENT STACK?

The term is quite liberally used online, but generally speaking, web development stack is a set of tools typically used in tandem to develop web apps. A stack can (and is meant to) be used repeatedly to develop web applications. Individual developers as well as software companies often specialize in a specific stack or several stacks.

As a potential client of a software company or someone interested in developing a new web or mobile app, you should definitely be interested in the web development stack of the company you wish to work with. A highly specialized web stack can exist within the constraints of frontend or backend development. That's why we can talk about:

V. FRONTEND WEB DEVELOPMENT STACKS

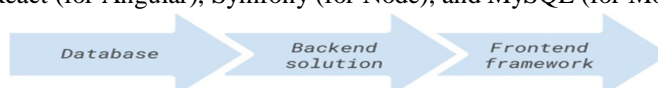
Loosely defined set of frontend technologies often used together to make an app. They may include such types of software as frontend frameworks and libraries, package managers (used among others to quickly install all the dependencies of a given piece of software), build systems (to automate various tasks), testing tools, version control systems, caching tools, or deployment software. At TSH, frontend developers develop highly specialized frontend stacks that include many other low-level tools that work best when used together. There is no need to memorize them but if you are interested in this topic, you can always ask them yourself.

VI. BACKEND WEB DEVELOPMENT STACKS

A collection of backend tools and technologies that developers tend to use in tandem. Much like with frontend, they can include many classes of software, including containerisation tools (e.g. Docker – you can use it to deploy your app more easily and scale it up), APIs (for the UI and developers to communicate with the system), databases, search engines (e.g. Elasticsearch), caching mechanisms or even DevOps tools.

A. Full Web Development Stacks

The details behind both frontend and backend stacks can very well be left for developers to worry about. From a strategic point of view, the most important choice is the one concerning your full development stack. That's what we're going to talk about from now on. Refers to web technology sets that include all the essential parts of a modern app: the frontend framework, the backend solution and the database. These three are (most of the time) obligatory, but a stack can also sometimes include other elements, such as an application framework (e.g. Express for Node), or state management tool (e.g. Redux often used with React, which requires the help in large apps for better scalability). Sometimes, full web development stacks are abbreviated based on the technologies that comprise them. A good example would MEAN (Mongo, Express, Angular, Node), but any combination of technologies could potentially work together, regardless of whether they have a fancy abbreviation. For example, the three obligatory ingredients in the mix could be replaced by Vue or React (for Angular), Symfony (for Node), and MySQL (for MongoDB).



Typical essential ingredients of a full-stack in web development

B. Full Web Development Stacks

Before we go over specific stacks popular in 2021, let's find out more about the building blocks of the stack.

C. The Frontend Framework

First, let's start with something a little counterintuitive. Many say that just because something is popular, it doesn't mean that it's good. While technically true, when it comes to frontend frameworks, popularity is usually the result of how good a given technology is. The choice on the frontend is somewhat limited as there is only one programming language – JavaScript (which has nothing to do with Java!). With that in mind, all we need to do is pick a framework. Some of the most popular frontend/application frameworks are Angular, React and Vue. All the other low-level choices are tied to the big choice – for example, if you choose React, you are likely to also use Redux for state management in your app. But that issue can be left to the programmers.

For example, using React Native makes it possible to share a lot of code (and save some time and effort in the process) between a React-based web app and a React Native-based mobile app

– watch our latest webinar that covers React development for both web and mobile:

VII. THE BACKEND SOLUTION

Just like in the case of frontend frameworks, popularity is a good measure of the solution's viability. As a general rule, the more popular it is, the better support (e.g. helpful community, rich documentation, third-party extensions and tools) it offers. However, be wary of sudden spikes of popularity as factors in your decision. Once you choose a backend solution, it's bound to stay with you for a long time – it is quite costly to convert to an entirely different language.

Speaking of programming languages, it's worth it to know that today choosing a programming language is redundant. That's because today pretty much all developers work with frameworks and the choice of a framework also determines the language. For example, Symfony or Laravel are written in PHP, Express is tied with Node, while Django or Flask entails the choice of Python as your backend language. When you already have a web app or website and reach out to a software company, make sure you can tell them what framework (including content management systems) was used to make it.

Node is a backend solution that needs some more explanation. It's not a language on its own, but rather it is a way to execute JavaScript, the frontend language, on the backend. When you choose Node, you can have an application that uses JavaScript as its only programming language. While there are many differences between using this language on the frontend and backend, the similarities are enough to ensure a large number of full-stack developers who did both.

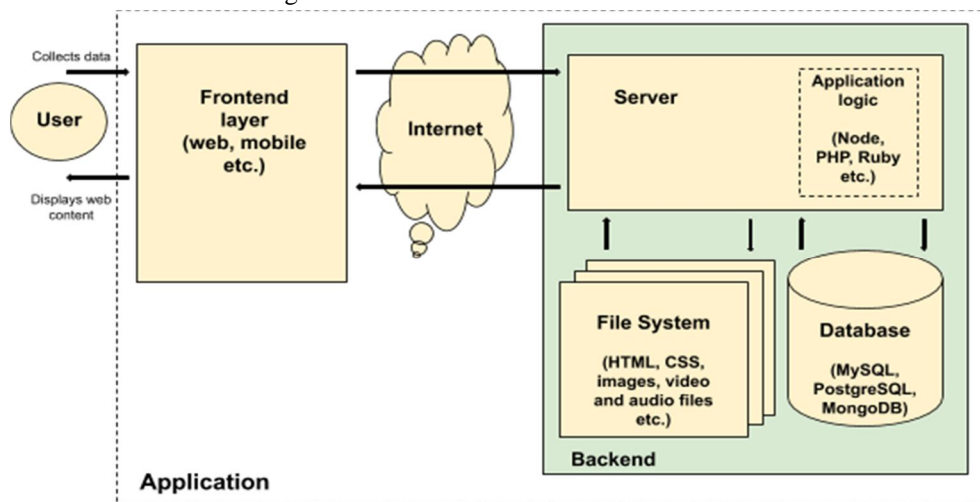
A. The Database

Every modern web or mobile application requires a database. The truth is that whichever specific database you choose is not that important. What you should know is whether it is:

- 1) a relational database (such as SQL),
- 2) or document-oriented database.

Relational databases tend to keep all the data in a very structuralized way, in which various data types are related to each other. Document-oriented databases offer a more loose structure. Some will argue strongly in favor of one of the solutions, but it largely depends on the kind of app you want to develop. If its data types naturally fall into easy to predict structures and relations, a relational database will be a better option if you don't want to end up with a mess. On the other hand, if a large part of your data is highly unique and impossible to relate to each other, a relational database may require lots of tweaking to work well.

As your app gets more and more popular, your database may often end up being the single most valuable asset you have. You should pay a lot of attention to this issue right from the start.



Simplified relationship between different parts of the web application

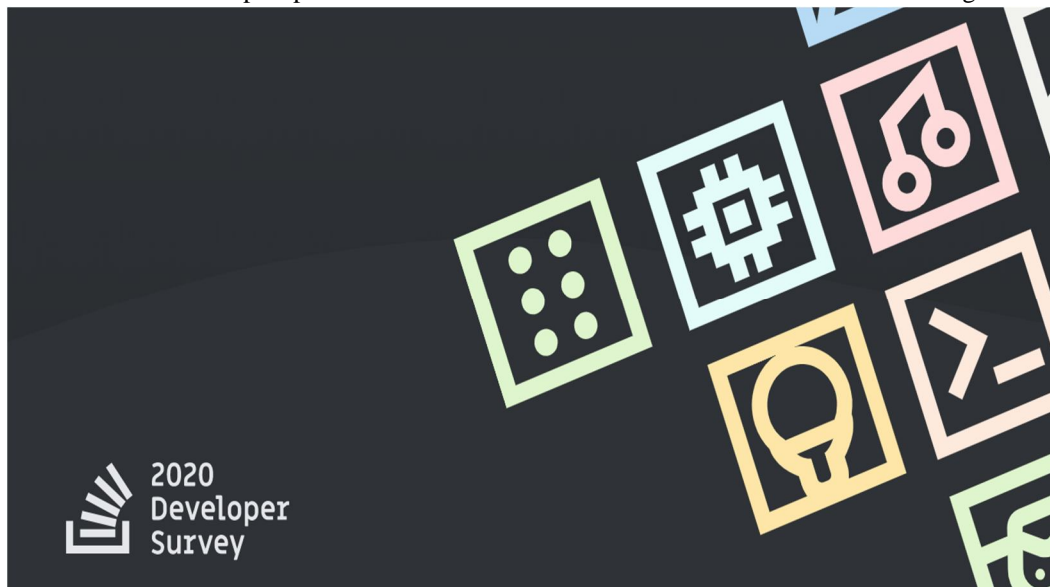
B. Web Development Stacks in 2021

So what web development stacks are the most popular in 2021? A good source of this information is the latest Stack Overflow Developer Survey.

When it comes to frontend frameworks, React comes out on top (well, technically it is jQuery, but it is more of a library rather than a full framework and it can be used together with any framework), with Angular and Vue not too far behind.

JavaScript is the most popular technology overall with as many as 67,7% of all respondents admitting to using it to some extent. Considering the popularity of Node and the fact that it has no competition on the frontend, it is quite predictable. Python, Java and PHP are also all relatively popular. It's worth it to note that TypeScript (used by 25.4% of all respondents) is yet another proof of JavaScript's popularity, as it is in fact a superset of JavaScript, adding some optional new features to the language. Any JavaScript program is also a valid TypeScript program.

What about databases? According to the survey, MySQL, PostgreSQL, Microsoft SQL Server, SQLite and MongoDB are the most popular. The annual Stack Overflow report provides a wealth of information on the trendiest web technologies



C. What is a full-stack developer? Are they still a thing in 2021?

Quite a lot of information, isn't it? You might now wonder: "Wait, there are supposedly those 'full-stack developers'. Does it mean that they can do all that described above?". Well, not exactly.

Back in the early days of the web, when the apps were not so big and complex and the software development industry far less specialized, it was more than possible to be a full-tack developer with some knowledge of CSS, JavaScript, jQuery and PHP.

These days full-stack development should be thought of in a little different way. They usually are experienced devs who worked as both frontend and backend developers and they have enough knowledge of both to use it to the benefit of the team, making it easier for the frontend and backend team to communicate with each other. Such devs are very valuable and at TSH they are often made tech leads or senior developers.

There are also individual freelance developers who take on whole projects by themselves. But this is only an option for relatively simple web development tasks, such as CMS-based websites and basic apps that can be made mostly from ready-made software. Large custom web applications are simply too complex and require too great a variety of skills and specializations for one person to develop in a reasonable amount of time.

D. How to Choose the Right Web Development Stack?

The choice will ultimately be the result of discussions with your developers or the software company you're going to work with, but there are some things you can do improve your chances of getting it right:

- 1) Make sure to pick the right software company. Not sure how? Read this article on how to choose a software company.
- 2) If you made your research diligently, you now have a partner you can discuss this topic with and follow their advice. If you already have an app, take their advice into consideration as well. Chances are that your stack may be in need of an update for various reasons, including scalability and technologies getting obsolete.
- 3) Be informed at every stage of the process as much as you can – by improving your methodology of choosing the right developers and vendors, learning more about stacks and asking for advice. CTO of the software company you partner with should be a great source of high-level strategic information about web development stacks.

What about stacks The Software House? What programming languages and other tools do we specialize in? Our developers mostly tend to specialize in either JavaScript or PHP web development services. That's why we use:

- a) React, Vue_or Angular_on the frontend.
- b) Node/JavaScript_as well as PHP's Symfony_and Laravel_as the main backend solution.
- c) React Native, Android_and iOS_native technologies as well as Flutter_for mobile development.
- d) PostgreSQL, MySQL or Mongo as databases.
- e) Various low-level frontend and backend stacks centered around React (including TSH's own React starter for frontend toolset) or Docker (backend).

And many more! Want to learn more about what kind of technologies we use commonly, sometimes, only as an experiment, or try to avoid completely? Check our TSH Technology Radar. And if you have doubts about any of the technologies included in the radar, make sure to contact us!

VIII. ADVANTAGES OF A MERN STACK

A. UI Rendering and Performance

React JS is the best when it is about UI layer abstraction. Since React is only a library, it provides you the freedom to build the application and organize the code however you want. So, it is better than Angular in terms of UI rendering and performance.

B. Cost-Effective

As MERN Stack uses one language throughout that is Javascript so it will be beneficial for a company to hire Javascript experts only rather than hiring different specialists for different technology. This move will save a lot of time and money.

C. Open Source

All technologies that are involved in MERN are open-source. This feature allows a developer to get solutions to queries that may evolve during development, from the available open portals. As a result, it will be beneficial for a developer.

D. Easy to Switch Between Client and Server

As everything is written in one language this is why MERN is simple and fast. And also it is easy to switch between client and server.

IX. DISADVANTAGES OF MERN STACK!

A. Productivity

Since React is just a library it uses many third-party libraries which provides lower developer productivity. And due to this upgrading, the React code requires more effort.

B. Large-Scale Applications

It becomes difficult with MERN to make a large project in which many developers are involved. MERN stack is best suited for single-page applications.

C. Errorprevention

If you want a technology stack that prevents common coding errors by its very design, then the MEAN stack is a better choice. As Angular uses Typescript, so prevents common coding errors at the coding stage itself. However, React lags behind here.

X. MERN STACK FOR DEVELOPERS

The MERN stack's main advantage in web development is that each line of code is written in JavaScript, which is a nearly universal programming language since it is vital for both server-side and client-side code. By using a single programming language, the MERN stack eliminates the need for context switching and greatly simplifies the entire development process, giving web developers the tools to create efficient web applications with far less effort. Technology stacks that use multiple programming languages leave developers to figure out for themselves how to properly interface them, but by using a JavaScript stack like MERN, developers only need to know JavaScript and JSON. To become a Web developer you can join [MERN Stack online training class](#).

A. Importance And Future Scope Of Mern Stack

Working in the MERN stack delivers powerful results simply and efficiently. Knowing your way around the stack is an important skill set since building and deploying solid MERN apps is likely to gain importance in the future.

These days, MERN Stack is used extensively since everything is done in JavaScript. As you know, JavaScript is everywhere. It is used both on the front-end and back-end side. Because of this, there's no need for context switching.

Tech stack that utilize multiple coding languages, force developers to figure out how to mix them together. Since MRN is JavaScript-based, developers only need to master a single coding language, which makes things a million times easier.

Node.js was built on Chrome's JavaScript runtime to make it more conducive to building fast- operating network applications with easy scalability. The platform operates using a no blocking, event-driven I/O model that is incredibly efficient.

Node.js's efficiency and simplicity make it an ideal platform for real-time applications running across distributed devices, especially those with intense data requirements. But Node.js has far more potential in conjunction with the MERN stack.

B. Features of MongoDB

- 1) Has Document-Oriented storage.
- 2) Has a single master with built-in replication support.
- 3) Ships with built-in sharding support (*spreading data around several replicated clusters*).
- 4) It has very wide driver support, although databases ship with limited support, MongoDB has drivers for every language.
- 5) Provide horizontal scalability to balance the increasing load of modern applications.
- 6) Robust, flexible, and scale-able.

XI. CONCLUSION

So, we can say that MERN Stack has bright future ahead. In order to get started with MERN Stack development, all you really need is a good understanding of JavaScript and ES6 Fundamentals, but also at least some familiarity with the basic concepts of React and Node.js. There are many MERN Stack online training that will have you building your own dynamic web applications in no time. All you really have to do is pick the one that seems best to you and get started right away!

REFERENCES

- [1] "us-en_cloud_learn_mean-stack-explained". www.ibm.com. IBM Cloud Education. 2019-05-09. Archived from the original on 2020-05-14. Retrieved 2020-02-16.
- [2] Dickey, Jeff (2014-09-24). Write Modern Web Apps with the MEAN Stack: Mongo, Express, AngularJS, and Node.js. Peachpit Press. ISBN_9780133962376.
- [3] "LAMP vs MEAN, Deciding the right stack for your startup". www.linkedin.com. Retrieved 2020-02-16.
- [4] "The MEAN Stack: MongoDB, ExpressJS, Angular and Node.js". Tumblr. Apr 30, 2013.
- [5] "Mean Stack". LinkedIn.
- [6] "The most popular database for modern apps". MongoDB. Retrieved 2020-02-16
- [7] <https://www.sovereignconsult.com/blog/full-stack-development-what-is-mern-stack-and-its-advantages/>



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)