Top 25 Node.js API Frameworks For The Year 2022

Understanding the many frameworks Node.js has helped create can sound like a daunting task. It remains one of the most important to use one that works best for your organization of the project you want to work on. This article discusses the best 25 Node.js Frameworks that you must know of.

Written by:- Chintan Gor

Table of contents [show]

The IT sector is one of the fast-paced and ever-changing industries across the world. Technology requires a flexible platform that can easily pacify these rapid changes to produce diverse applications with ease. The need gave rise to the development of the Node JS Frameworks!

Currently, Node JS is the fastest server-side web application platform because it gives Leading Node.js App Development Company the ease to build scalable, single program application development.

Progressing into this article will help you find more such details regarding it.

Why is Node.js Special?

- Its codebase is more consistent and cleaner.
- It builds enterprise applications faster that is also highly scalable.
- It is great for agile development as well as prototyping.
- It has a vast ecosystem that provides open-source libraries.
- The best of all, it uses the JavaScript language.

Architecture of Node

A good project architecture remains vital to create projects that can withstand future changes. The Node JS has a single-threaded event loop architecture ; this allows it to tackle the multiple concurrent demands with ease.

It does not follow the traditional "multithreaded request-response" model which uses multiple threads to handle the simultaneous client requests. This type of model generally makes complex and unpredictable programs that increase difficulties for dedicated software developers.

Its JavaScript event-based model comes with a JavaScript call back system as well. These features together form a versatile architecture and a developer's delight.

25 Leading Node.js REST API Frameworks

Express.js (GitHub Stars: 41,036)

It is one of the best Node JS API Frameworks that was built by TJ Holowaychuk. Express JS framework permits developers to develop enterprise applications that are single or double-paged and even hybrids.

These are an amalgamation of features that make the final product, whose main priority is to be highly scalable and speed.

Features:-

- It accompanies a model view controller feature that helps build eccentric apps.
- It acts like a routing table or set of routing libraries.
- It supports 14+ engine templates, HTTP methods, and a synchronicity for improved performance.
- It allows injection of heavy load packages that help in extending the functionality of the framework.

Pros:-

- The learning curve is low
- Perfect for development of a short app cycle
- Allows customizations
- Flexible and adjustable

Cons:-

Example:-

```
const express = require('express')

const app = express()

app.get('/', function (req, res) {

  res.send('Hello World')

})

app.listen(3000)
```

Hapi.js (GitHub Stars: 11,900)

It is one of the Node JS REST API Frameworks that assists intermediation between the server and client-side. It is considered as an excellent substitute for Express.js.

Hapi.js was modelled to master server functions. Also, it is an open source framework and

Features:-

It can give dynamic content as it arranges for a JavaScript template engine quickly.

It has a scalable architecture that is plugin based. These plugins allow building of real time chatting apps.

It has detailed API references as well as reliable support for the generation of documents.

Pros:-

Architecture is plugin-based

It is configuration centric

Web server is functionality rich

Supports microservices

Robust libraries

Cons:-

Endpoints need creating and tested manually

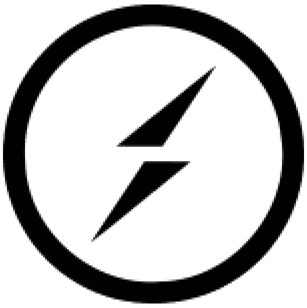
Refactoring too will happen manually

Example:-

```
const Hapi = require('@hapi/hapi');
```

```
  const server = Hapi.server({
      port: 3000,
      host: 'localhost'
  });
  await server.start();
  console.log('Server running on %s', server.info.uri);
};
init();
```

Socket.io (GitHub Stars: 48,200)



Socket.io is one of the Node.js Frameworks that is utilized to produce real-time computing apps. This framework is famous amongst developers because it gives accuracy and hastens the process.

Features:-

- It is exceedingly amicable and assists in auto-correction and anomaly detection, etc.
- It helps developers build a bidirectional flow of data between the server and the web.
- It aids developers in feeding clear streams to multiple browsers from the server.

Pros:-

- Real-time data analytics
- Socket authentication is smooth
- Assist in multiplexing
- Renders JSON serializable objects
- Customization of URL routing for web Sockets
- Fast

Cons:-

- Keeps connection open on a server
- Cannot create web applications that require periodic updates

Example:-

```
const server = require('http').createServer();

const io = require('socket.io')(server);
```

```
  client.on('event', data => { /* … */ });

  client.on('disconnect', () => { /* … */ });

});

server.listen(3000);
```

NestJS (GitHub Stars: 22,900)

This framework is flexible and can be used to build server-side applications that are streamlined and reliable. It makes use of progressive JavaScript, which is written on TypeScript.

The framework is accompanied by a set of strong typing models such as the Object Oriented Programming, Functional Reactive Programming and Functional Programming.

Features:-

- It builds healthy front-end development on AngularJS.
- Modular form of NestJS is such that it arranges code methodically in different modules.
- It can be used to build web applications that consist of features that are different

- Scalable technologies
- The ecosystem is vibrant with open source libraries
- Strong corporate support
- Event-based and fast

Cons:-

- Shows design issues
- Tool immaturity
- Congested performance

Example:-

```
import { NestFactory } from '@nestjs/core';

import { AppModule } from './app.module';

async function bootstrap() {

  const app = await NestFactory.create(AppModule);

  app.setViewEngine('hbs');

  await app.listen(3000);

}
```

```
bootstrap();
```

Salis.Js (GitHub stars: 21,000)

This is one of the Node.js Frameworks created on Express and hence offers very similar features. It uses code generators that will allow the building of applications with a lot less code written. Since it is highly compatible, you can easily use it to make browser-based applications.

Features:-

- HTTP request handled by the framework using Express.Js. These ensure superior quality.
- It is exceedingly compatible with websites that build frontend platforms.
- They can make smooth transitions as the MVC (Model View Controller) structure used is the same as many other frameworks.
- It gives users any data required; this becomes possible as it comes packed with ORM (Object Relational Mapping) solution.

Pros:-

- Data can be stored anywhere
- The separate user interface from Business logic
- Simple project templates

Cons:-

- Framework is slow
- Project development is slow
- Asset management is weak

Example:-

```
$ npm install sails -g

$ sails new test-project

$ cd test-project

$ sails lift
```

Meteor.Js (GitHub Stars: 40,490)

language like JavaScript. Hence, the applications you create will be treated as real-time applications.

To know Meteor in detail, analyze the Facts About Node JS Development. It will help you all

Features:-

- Using this framework will ensure that you do not have to renovate mobile applications.
- Js architecture is lightweight. This manages to reduce the code lines.
- It allows for smooth integration with Mango, React and Vue, etc.

Pros:-

- Syncing of catchphrases
- Architecture is Lightweight
- Abundant libraries and packages
- Robust community

Cons:-

- Lacks widget library
- No network connection flexibility
- Cannot make progressive web applications

Example:-

```
meteor create try-meteor

cd try-meteor

meteor
```

Koa.Js (GitHub Stars: 28,000)

It is a Node.js framework that was formed by the makers of Express.js. It is a small yet robust and expressive framework for the web.

Developers can use it to create web apps as well as API. It does not pack middleware inside of the centre. Hence, they are streamlined.

There is an ongoing battle among Express and Koa . You should look at it in detail.

Features:-

It provides extra customization; this allows developers to work on applications from the very beginning.

Its generators can limit the use of callbacks, which offer developers a better error handling system.

Pros:-

- Lightweight
- Supports sync/await keywords
- Generator functions transcend

Cons:-

- Its support community is smaller than most
- Compatibility with express style middleware is negligible

Example:-

```
const Koa = require('koa');

const app = new Koa();

app.use(async ctx => {

  ctx.body = 'Hello World';

});

app.listen(3000);
```

It is an influential and well-known Node JS REST API Frameworks as it ranks amongst the most popular ones. Why? Because it has an untroublesome CLI and a zestful API website browser. It will permit developers to construct their models.

Features:-

- A developer can quickly generate SDKs as well as API documents using this framework.
- It can easily connect to any database such as SQL, Oracle, MongoDB, etc.
- It can support numerous databases is its most outstanding feature.

Pros:-

- Full-stack
- Quick development
- Structured and modular code
- Models and features built-in

Cons:-

Example:-

```
import {get} from '@loopback/rest';

export class HelloController {

  @get('/hello')

  hello(): string {

    return 'Hello world!';

  }

}
```

Derby.Js (GitHub Stars: 4,500)

Derby.Js is one of the Node.js Frameworks that relies on MVC structure. This structure will

Features:-

It has an inbuilt rapid Derby Racer that acts as a data syncing engine. By allowing synchronization amongst server, app database as well as browsers flawlessly.

You can make real-time web apps by adding codes that are customized. The site hence produced will be customized.

Script development is bi-directional, which ultimately helps develop the client-side and the server-side.

Pros:-

Automatic conflict resolution

Custom web applications

Script same on server-side as well as client-side

Provides multiple editing options

Cons:-

Smaller community

Offers less support

Example:-

```
$ cd ~/derby-examples/directory

$ node server.js
$ cd ~/derby-examples/sink

$ coffee server.coffee

cd ~/derby-examples

node server.js
```

Total.js (GitHub Stars: 4,000)

This framework requires little to no assistance and is extremely quick. It promises to provide stable services along with the flexibility to construct IoT applications. You can also choose to build other desktop or web-based applications.

Features:-

- A NoSQL database was embedded on the Total.js and is more efficient in developing projects.
- It features a modular coding structure and will provide features like Simple Mail Transfer Protocol.
- It is compatible with diverse database systems such as MongoDB, Angular, MySQL, Ember, etc. this will allow front end development.

- Quick prototyping
- Can make real-time responsive applications
- Has predeveloped components
- Community chat

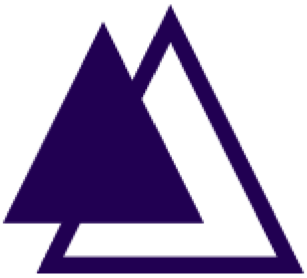
Cons:-

- Lacks a community
- Lacks support

Example:-

```
$ npm install total.js

$ cd workers

$ node index.js
```

Adnois JS (GitHub Stars: 7,400)



scratch. The ecosystem provided is such that it will help developers write stable server-side apps.

Features:-

- It offers consistent ecosystems that help developers by saving time of multiple package selection.
- JS supports the ORM method, thus permits exploring together with data handling.
- Allows swift work on current apps in case a developer faces troubles when utilizing this framework.

Pros:-

- Undemanding and improved
- Data handling is seamless
- Ecosystem is stable
- SQL queries are effective

Cons:-

- No configuration
- The community is not explorable

Example:-

```
$ npm i -g @adonisjs/cli
```

Mojito (GitHub Stars: 1,600)

It is one of the Node JS API Frameworks that dig out the kind of calling device as well as give out suitable HTML markup. Its large scale API modules can make REST calls, carry out code, handle cookies and assets, etc. Node.Js community hasn't yet embraced it yet.

Features:-

- It expands functionality by utilizing Y.Base.
- Since it's written in JavaScript, it can handle components on the server-side as well as client-side smoothly.
- It offers quick development of web applications.
- They allow for continuous growth while the application develops.

Pros:-

- Quick app development
- Allows leveraging utilities of both MVC framework
- Has server/client runtimes
- Widely used across Yahoo

Based on Express

Example:-

```
$ npm install -g mojito-cli

$ mojito create app hello

$ cd hello

$ mojito create mojit HelloMojit

$ mojito start

$ mojito test app

$ mojito docs app hello
```

Keystone.Js (GitHub Stars: 1,000)

as well as API's that are database driven.

Features:-

- It aids with multiple other apps and frameworks, for example- WCF, NET, ASP, etc.
- Js offers integration with online platforms such as Mandrill, Embedly, Google, etc. for easy handling data.
- It provides stable management of the progress in development, regardless of the operating system.

Pros:-

- Easy handling and customization
- Seamless integration with online platforms
- Highly extensible
- Fast on V8

Cons:-

- Packaging external libraries is difficult
- The default option to add more pages is absent
- No built-in roles system

Example:-

```
yarn create keystone-app my-app
yarn start
```

Feather.Js (GitHub Stars: 12,479)

This is one of the Node JS REST API Frameworks. Just like its name, it is lightweight. It can easily interact with the backend technology and support numerous databases.

The framework can also manage to work with frontend tech like- Reactive, Native, etc. It primarily behaves like a library that creates applications according to REST principles.

Features:-

- It serves as a real-time framework and has features that simplify development of modern web applications.
- They, at the same time, allow connecting with third-party services and applications.
- Even though it was developed as a microservice framework. It can help create minimalistic web frameworks.

Pros:-

- Universal usage
- Automatically supplies REST API's
- Can support rational along with non-rational database
- Error handling acceleration

Can’t work with sites that want to render views on the server

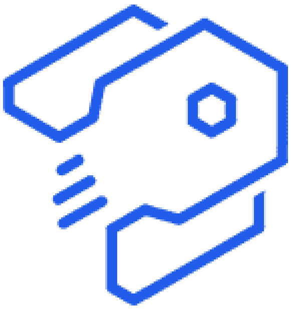
Takes longer time to learn

Example:-

```
const feathers = require('@feathersjs/feathers');

const express = require('@feathersjs/express');

const socketio = require('@feathersjs/socketio');

const memory = require('feathers-memory');

// Creates an Express compatible Feathers application

const app = express(feathers());

// Parse HTTP JSON bodies

app.use(express.json());

// Parse URL-encoded params

app.use(express.urlencoded({ extended: true }));
app.configure(express.rest());
// Configure Socket.io real-time APIs
app.configure(socketio());
// Register a messages service with pagination
app.use('/messages', memory({
  paginate: {
    default: 10,
    max: 25
  }
}));
// Register a nicer error handler than the default Express on
app.use(express.errorHandler());
// Add any new real-time connection to the `everybody` channe
```

```
  // Publish all events to the `everybody` channel

  app.publish(data => app.channel('everybody'));

  // Start the server

  app.listen(3030).on('listening', () =>

    console.log('Feathers server listening on localhost:3030')

  );
```

Strapi.io (GitHub Stars: 24,900)



It is an open-sourced framework that possesses the ability to govern the content and allocate it anywhere. You can easily customize the admin panel and the API as well. The focus of this framework is to provide systematic, well planned, and fast applications and websites.

Features:-

- Every project will have inbuilt features such as- authentication, authorization management, and API generators.
- It is heavily customizable and will enable developers to make their plugins.
- It can be used as a microservice, thus offering infinite possibilities to stitch the applications.

Pros:-

- Customizable
- Self-hosted
- Keeps control of the data
- Easy setup
- Large community

Cons:-

- Internalization features missing
- CLI not working

Example:-

```
yarn create strapi-app my-project --quickstart

yarn develop
```

Restify.Js (GitHub Stars: 9,800)

It is one of the well-known Node JS REST API Frameworks that optimizes introspection and performance—often used by some well-known companies such as Netflix, Pinterest, and Napster. Its' framework has been called the future of Node.js Rest development.

Features:-

- Purely made to build scalable APIs.
- Seamlessly works in harmony with HTTP actions, response and cycle requests.
- It comes in handy when working on DTrace by offering automatic support.
- This framework can maintain real-time data and also offers bi-directional communication.

Pros:-

- Debuggable
- Production ready
- Semantically correct

Cons:-

- Cannot create regular web projects

Example:-

```
function respond(req, res, next) {
  res.send('hello ' + req.params.name);
  next();
}
const server = restify.createServer();
server.get('/hello/:name', respond);
server.head('/hello/:name', respond);
server.listen(8080, function() {
  console.log('%s listening at %s', server.name, server.url);
});
```

ActionHero.Js (GitHub Stars: 2,100)

It acts as a framework that possesses multi-transportable API along with non-segregated cluster capabilities. Its main aim is to make easy toolkits ensuring reusable & scalable API services.

The framework has more features to offer such as flexibility which usually varies from app to app.

Features:-

- Its cache system is top-notch, and so is its communication, and decentralization features.
- It runs similar API's through numerous protocols.
- This framework is cluster ready with inbuilt assistance for background tasks.

Pros:-

- Scalable
- Same API for numerous transports
- Lightweight
- Simple

Cons:-

- 0-downtime deployment

```
npx actionhero generate

npm install

npm run build # <--- new! I compile the TS to JS

npm run dev # <--- new! I use `ts-node` to let you develop on

# Use the actionhero CLI

(npx) actionhero generate action --name my_action

(npx) actionhero generate task --name my_task --queue default

# Test

npm test

# I'll run `prettier` and `build` for you first

# I handle .ts files now!

# To deploy your app
```

Diet.js (GitHub Stars: 379)

It is one of the compact modular Node JS Frameworks which constructs fast and scalable applications. The host controller uses the server one or more hostnames. Also, it is MIT licensed! You should know about it as Node JS is the Next Model of Development .

Features:-

- Its size is about 450 sloc which means it is the smallest.
- It can write clean API's in addition to URLs from scratch making them customized.
- The framework allows semantic and natural scattering by app.post and also other processes.

Pros:-

- Middleware assistance
- MVC assistance
- Virtual hosting

Cons:-

Example:-

```
var server = require('diet')

var app = server()

app.listen(8000)

app.get('/', function($){

    $.end('hello world')

})
```

Mean.js (GitHub Stars: 12,000)

Mean is the acronym for Angular 6, Node, Express and Mango. These make it a technology of open sources. It allows developers to develop applications that are powerful, fast, and manageable. So, Developer Choose Mean Stack JS a lot nowadays.

Since it is a development stack, developers have no worries about additional components.

Reaps benefits of cloud compatibility by implanting functionalities.

Offers real-time changes of the applications even if they are in the development stage.

Pros:-

Same language used in backend and frontend

Easy to work

Flexible

Cons:-

Complicated

Needs working knowledge

Example:-

```
git clone https://github.com/linnovate/mean

cd mean

cp .env.example .env

yarn
```

yarn start (for development)

Sequelize (GitHub Stars: 21,900)

This web scraping node js frameworks acts as a generator framework. It has been called the promise-based ORM technique. It allows for the framework to support features such as association, validation, and synchronisation.

Features:-

- Many of its query-based models are like MongoDB, especially CRUD.
- It supports numerous databases which provide developers with immense flexibility.
- The errors texts are customizable because of implementations in the middleware.

Pros:-

- Model hooks
- Transaction support
- Data seeding

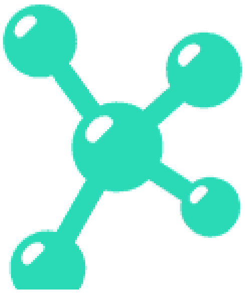
Cons:-

- Inefficient

Example:-

```
$ npm i sequelize # This will install v5

$ npm i sequelize@next # This will install v6-beta

# And one of the following:

$ npm i pg pg-hstore # Postgres

$ npm i mysql2

$ npm i mariadb

$ npm i sqlite3

$ npm i tedious # Microsoft SQL Server
```

Molecular (GitHub Stars: 3,500)



middlewares and has open-sourced libraries.

Features:-

- It is a fault-tolerant framework.
- It is extremely extensible and provides built-in modules that are customizable.
- Allows the building of apps for various platforms, be it web, mobile, or IoT

Pros:-

- Open-sourced
- Extensible
- Blazing fast
- Memory caching

Cons:-

- Complex
- Masks low level failures

Example:-

```
const { ServiceBroker } = require("moleculer");

// Create a broker
// Create a service

broker.createService({

    name: "math",

    actions: {

        add(ctx) {

            return Number(ctx.params.a) + Number(ctx.params.b

        }

    }

});

// Start broker

broker.start()

    // Call service

    .then(() => broker.call("math.add", { a: 5, b: 3 }))
    ).catch(err => console.error(`Error occurred: ${err.messag
```

Flatiron.Js (GitHub Stars: 1,300)

It has a packaged based approach, and that sets it apart from other Node JS Frameworks. It gives developers the liberty to incorporate as little or as much as they choose to. This framework can handle essential data management by supplying strong ODM.

Features:-

- Helps in faultless incorporation with online platforms such as Mandril, Cloudinary, google, etc.
- Enables smooth development of websites and apps by assisting in dynamic routes and processing.
- It provides consistent management toward development progress.

Pros:-

- Full-stack
- Async logging library
- Hybrid streaming
- Directs traffic with ease

Complex

No active development now

Example:-

```
var flatiron = require('flatiron'),

    app = flatiron.app;

app.use(flatiron.plugins.http);

app.router.get('/', function () {

  this.res.writeHead(200, { 'Content-Type': 'text/plain' });

  this.res.end('Hello world!n');

});

app.start(8080);
```

Read also: Node.js Development Trends For The Year 2020

FoalTS (GitHub Stars: 751)

It is majorly used to build HTTP APIs and apps with affluent interfaces. It has conveniently been written on JavaScript, which ensures the code is straightforward while maintaining smoothness. Also, it does not require you to make the code from scratch.

Features:-

- They are written with TypeScript, which supplies clean and concise code.
- Its architecture and other parts are designed to ensure the code is uncomplicated.

Pros:-

- Lightweight
- Testable code
- Progressive apps

Cons:-

- Code takes long to compile

```
$ npm install -g @foal/cli

$ foal createapp my-app

$ cd my-app

$ npm run develop
```

VulcanJS (GitHub Stars: 7,900)

This full-stack framework provides developers with tools to build React, and GraphQL established web apps. It can speed up the repetitive task of front and back end coding, and hence the result is exceptional.

Features:-

It handles client-side formation generated and their submissions using suitable Apollo mutation.

Uses Meteor as the backend layer that handles databases.

Provides data loading helpers and hence ensures easy loading of Apollo data.

- GraphQL Schema generation

Cons:-

- Small community
- Relies on standard libraries

Example:-

```
Package.describe({
  name: "my-custom-package"
});
Package.onUse( function(api) {
  api.versionsFrom("METEOR@1.0");
  api.use([
    'fourseven:scss',
    'vulcan:core',
    'vulcan:posts',
    'vulcan:users'
  ]);
   api.mainModule('server.js', 'server');
  api.mainModule('client.js', 'client');
  api.addFiles([
    'lib/stylesheets/custom.scss'
  ], ['client']);
  api.addAssets([
    'lib/server/emails/customNewPost.handlebars',
    'lib/server/emails/customEmail.handlebars'
  ], ['server']);
});
```

It is one of the feature-rich node js testing frameworks that makes asynchronous examination easy and enjoyable. The framework is a testing library for Node.js. It was created to be simple, fast, and extensible.

Features:-

- It is well-liked for being open-sourced, flexible, and providing custom assertion libraries.
- It seamlessly works with Browserstack to test websites and mobile apps while also reporting the test durations.
- It has a proper exit status for CL support and also detects any leaks globally.

Pros:-

- Test coverage reporting
- File watcher support
- Slow test highlighted

Cons:-

- Complex

Example:-

```
<html>

<head>

  <!-- add mocha css, to show results -->

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/a

  <!-- add mocha framework code -->

  <script src="https://cdnjs.cloudflare.com/ajax/libs/mocha/3

  <script>

    mocha.setup('bdd'); // minimal setup

  </script>

  <!-- add chai -->

  <script src="https://cdnjs.cloudflare.com/ajax/libs/chai/3.

  <script>
  </script>

</head>

<body>

  <script>

    function pow(x, n) {

      /* function code is to be written, empty now */

    }

  </script>

  <!-- the script with tests (describe, it...) -->

  <script src="test.js"></script>

  <!-- the element with id="mocha" will contain test results

  <div id="mocha"></div>
```

```
      mocha.run();

    </script>

  </body>

  </html>
```

Conclusion

These were the 25 of the best Node JS Frameworks. This article highlights details such as features and pros & cons of each of the 25 frameworks. They will help any NodeJS Developer to choose the one that suits the best for their business. Thank You.!

Frequently Asked Questions

What Is Node.js Framework?

What Is The Best Framework For Node.js?

There are a lot of options available for the Node.js framework. Some of the most popular ones are as listed below: Express.js Hapi.js Socket.io NestJs Sails.Js Meteor.Js

What Is The Difference Between Node.js & Framework?

What Are The Major Differences Between Node.js & PHP?

What Is The Most Popular MVC Framework Of Node.js?

What Could Be The Best Alternative For Node.js?

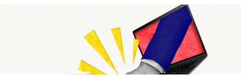
Chintan Gor

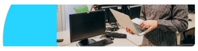
Enthusiastic for web app development, Chintan Gor has zeal in experimenting with his knowledge of Node.js in various aspects of development. He keeps on updating his technical know-how thus pinning his name among the topmost CTO's in India. His contribution is penned down by him through various blogs on Node.js. He is associated with eSparkBiz - a Node.js Development Company from the past 7-8 years where one can get premium services.

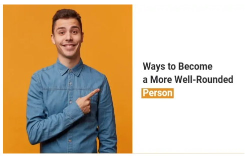
Recent Popular

Nearshore vs. In-house Software Development

Nearshore Software Development: The Definitive Guide for CEOs


Stages of a Successful Product Development Strategy for Ecommerce





Ways to Become a More Well-Rounded Person

Related Services

Fintech Development

Healthcare Development

Real Estate Development

Education Software Development

Need Help?

We are here for you

Step into a new land of opportunities and unearth the benefits of digital transformation.

Enter your name *

Enter your email Id *

Your number *

Hire Developer

Submit

Related Post

Node.Js June 10, 2021

Why Node.js is the Best Choice to Develop an E-commerce Website

This is the era where all sorts of businesses are shifting towards digital...

View All

Formed in 2010, eSparkBiz is a world-recognized software development company, offering businesses a culture of innovations. Our mobile and web development solutions provide businesses a competitive edge and help them overcome their businesses limitations.