


# 100% Job Guarantee In IT

- ✓ Government Registered Training Institute
- ✓ Current Industry Syllabus
- ✓ Industry-Ready Prep Course With 100% Job Guarantee
- ✓ 100% Project Oriented Training From Basics
- ✓ Unlimited Interviews And Internship Support
- ✓ Individual Focused Training
- ✓ Suits For Fresher, Experienced, Non-IT
- ✓ Check Our  YouTube Videos Before Choosing Us



## OUR TRAINEES GOT PLACED IN



.....Still  
Counting

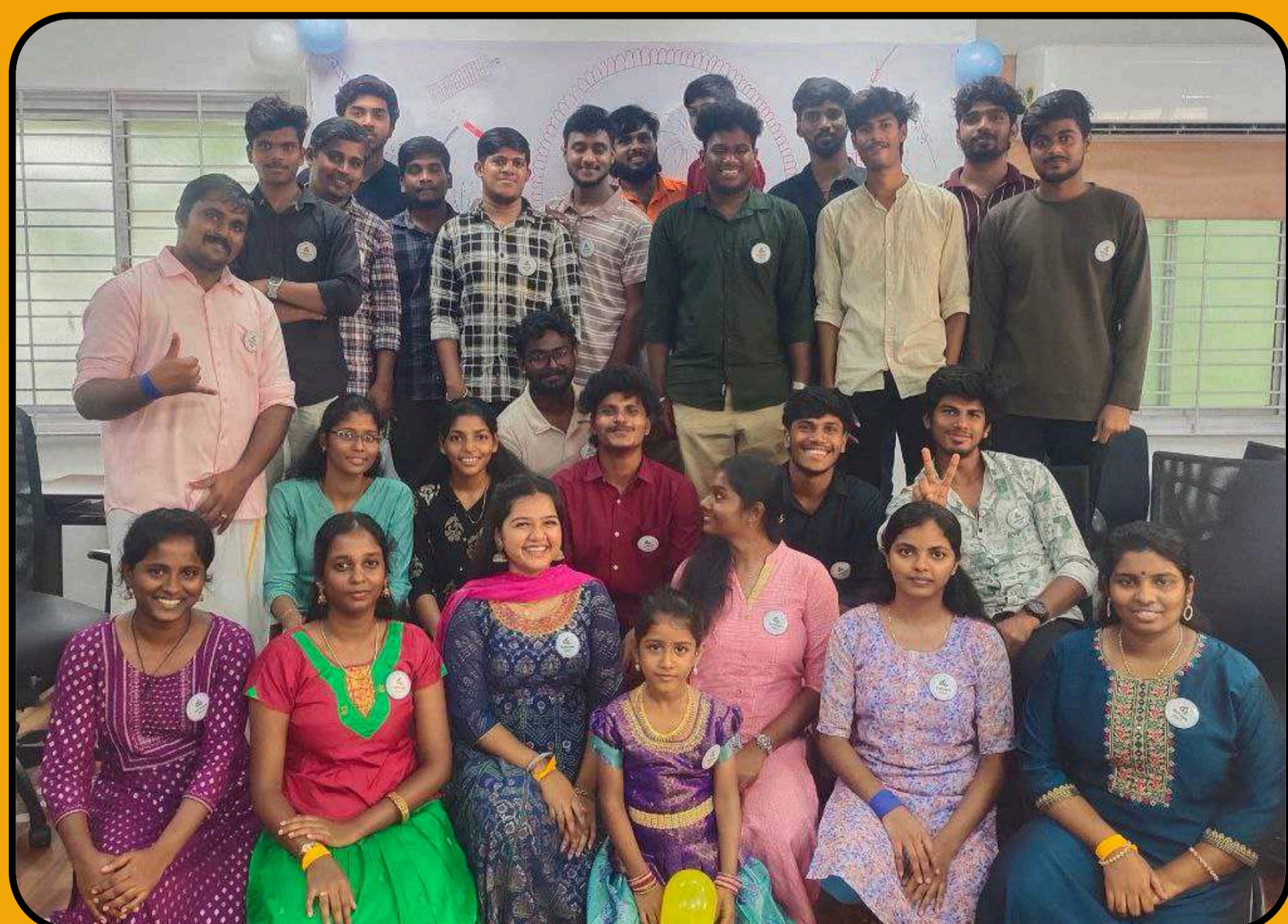


## ABOUT US!

Payilagam, the **Best Software Training Institute In Chennai With 100% Placement Guarantee**, has been shaping successful IT careers with over **13+ Years Of Experience**. we provide real-time, industry-focused training led by **Experienced Professionals** from the IT industry, ensuring you gain practical skills that companies actually need. Our courses include **HTML, CSS, JavaScript, React, Java, Python, PostgreSQL, GitLab, And WordPress**, designed to make you job-ready from day one. At payilagam, we don't just teach, we guide you step-by-step towards your career with our **100% Job Guaranteed Courses**, helping you confidently step into the it industry.



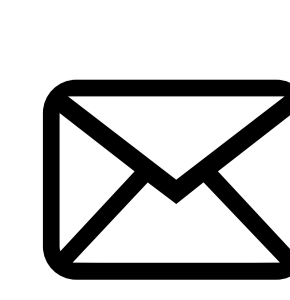
## OUR INFRASTRUCTURE



[www.payilagam.com](http://www.payilagam.com)



+91 8344 777 333



[info@payilagam.com](mailto:info@payilagam.com)

# OUR LATEST PLACEMENTS



Megha **BE AI (2025)**

ML DATA ASSOCIATE ROLE 1



Sasireka **MSc. CS (2025)**

ASSISTANT SYSTEM ENGINEER



Divya Bharathi **BE ECE (2026)**

SOFTWARE ENGINEER



Ramya **B.E. CSE (2025)**

DATA ANALYST



Vidhya Varshini  
**B E Aeronautical (2025)**

WEB DEVELOPER



Deenadhayalan **BSc CS (2025)**

TECHNICAL SUPPORT ENGINEER



Dharshini E **B.E CSE (2025)**

FULL STACK DEVELOPER INTERN



Vimala G **BE ECE (2016)**

SOFTWARE ENGINEER



Ramesh **B.Com (2025)**

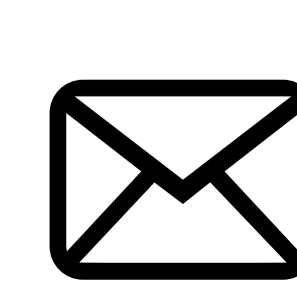
SOFTWARE ENGINEER



[www.payilagam.com](http://www.payilagam.com)



+91 8344 777 333



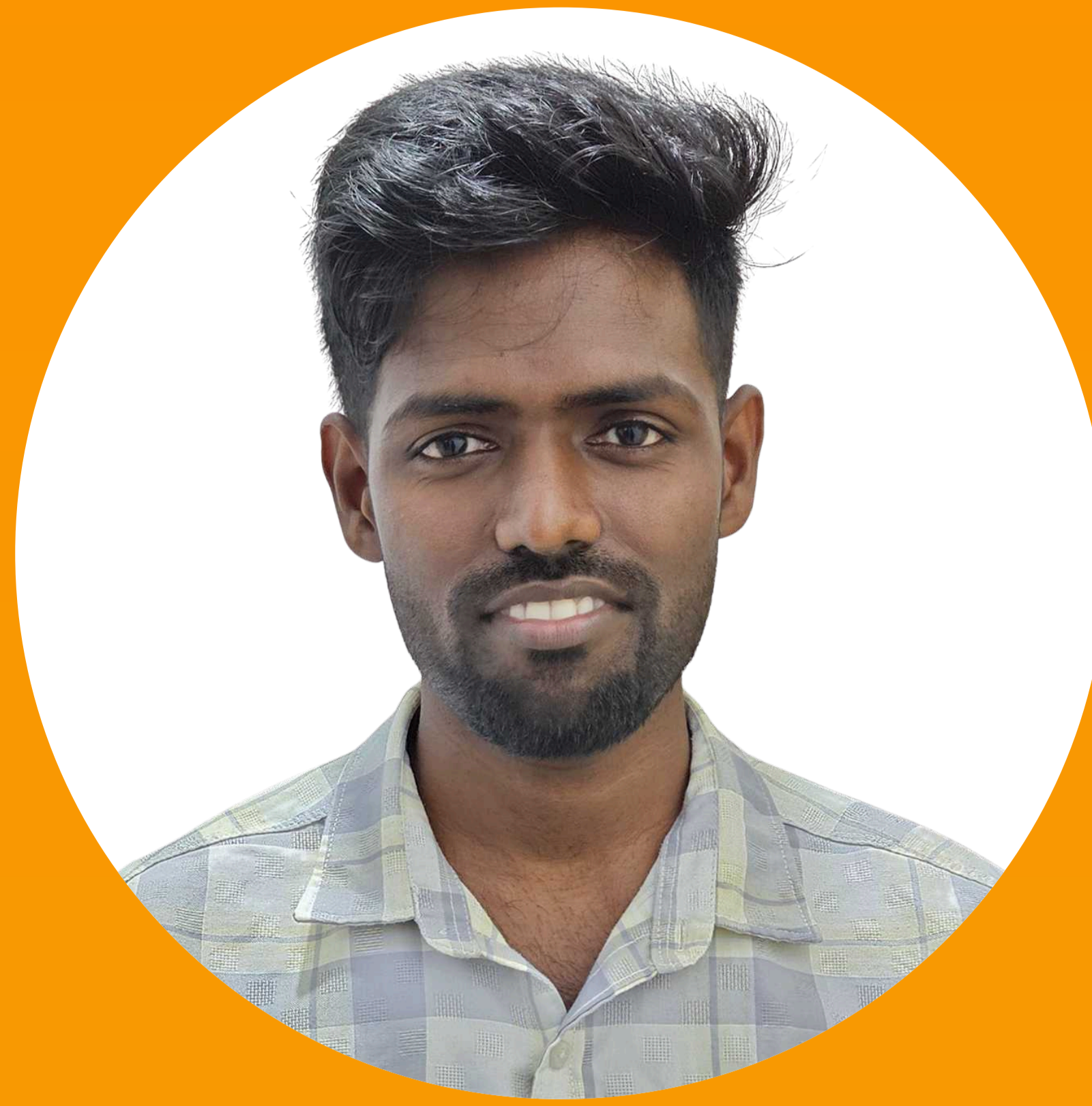
[info@payilagam.com](mailto:info@payilagam.com)

# OUR LATEST PLACEMENTS



Sathish **MSc (2023)**

SOFTWARE ENGINEER



Ranjith **BSc Physics(2021)**

ANGULAR DEVELOPER



Bharath **MSc Maths ( 2021)**

ANGULAR DEVELOPER



Iyappan **MCA (2025)**

SOFTWARE ENGINEER



Bhuvana Sri R **B.E CSE (2025)**

SOFTWARE ENGINEER



Pavithra **MCA (2025)**

NETWORK ENGINEER



Keerthika **MCA (2025)**

REACT DEVELOPER



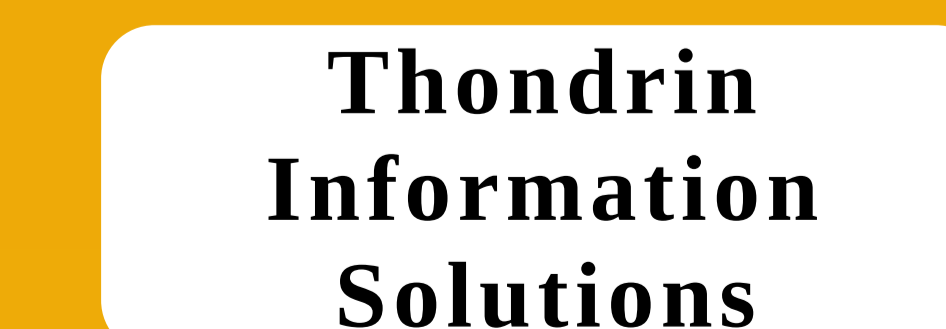
Vasanth S **B.E Mech ( 2021)**

SOFTWARE ENGINEER



Gayathri **BE CSE (2025)**

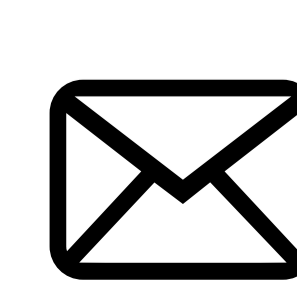
SOFTWARE ENGINEER



[www.payilagam.com](http://www.payilagam.com)



+91 8344 777 333



[info@payilagam.com](mailto:info@payilagam.com)

# OUR LATEST PLACEMENTS



Vadivu Lakshmi G  
**MSc Chemistry(2018)**

FULL STACK DEVELOPER INTERN



Ponvel M **BE CSE (2024)**

SOFTWARE ENGINEER

Thondrin  
Information  
Solutions



Uthaya **ME (2015)**

ANGULAR DEVELOPER



AJAY RAJA **B.Tech CSE (2025)**

DEVOPS ENGINEER INTERN



Ashok **BE ECE (2025)**

DATA ANALYST INTERN



Manikandan **B.Sc (2025)**

DATA ANALYST



Vigneshwaralingam  
**B.E CSE (2025)**

SOFTWARE ENGINEER



Swetha.P **B.Sc CS (2025)**

SOFTWARE ENGINEER INTERN



Kavya S **BE CSE (2025)**

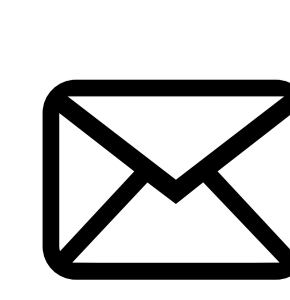
SOFTWARE ENGINEER INTERN



[www.payilagam.com](http://www.payilagam.com)



+91 8344 777 333



[info@payilagam.com](mailto:info@payilagam.com)

## HTML & CSS

### Project 1: Facebook / Instagram Login Page

- Introduction To HTML Syntax, Introduction To CSS
- Block-Level Vs Inline Elements
- Common Elements: <div>, <span>, <p>, <a>, <br>, <hr> & So On
- Flex Properties, Input Types, Nesting Elements Correctly
- Basic Selectors: Element, Class, ID
- Grouping And Combining Selectors, Attribute Selectors
- Combinators:Descendant, Child (>), Adjacent Sibling (+), Sibling (~)

### Project 2: Google Landing Page

- Global Attributes: Id, Class, Title, Style, Lang
- Ordered Lists: <Ol>, <Li>, Unordered Lists: <Ul>, <Li>
- Description Lists: <Dl>, <Dt>, <Dd>
- Data Specific Attributes: href, src, alt, type, value, placeholder
- Boolean Attributes: checked, disabled, readonly, required

### Project 3: Forms Using HTML & CSS

- <form> Element And Its Attributes (action, method, etc.)
- Input Fields: <input>, <textarea>, <select>, <option>, <button>
- Fieldsets, Legends, And Labels
- Form Validation (required, pattern, minlength, maxlength)
- Understanding Content, Padding, Border, And Margin
- Using Box-Sizing: Border-Box
- Visualizing Box Dimensions With Dev Tools, Margin Collapsing

### Project 4: Resume

- Creating Tables With <Table>, <Tr>, <Th>, <Td>
- Table Structure: <Thead>, <Tbody>, <Tfoot>
- Merging Cells With Colspan And Rowspan
- Table Accessibility And Semantics
- Nesting And Styling Lists
- Block, Inline, Inline-Block, None, Flex, Grid
- Visibility: Hidden Vs Display: None



## HTML & CSS

### Project 5: YouTube

- Introduction To Grid Layout
- Creating Columns And Rows Using Grid-Template-Columns And Grid-Template-Rows
- Placing Items With Grid-Column, Grid-Row
- Grid Gap, Alignment, And Nested Grids

### Project 6: Product Card

- Static, Relative, Absolute, Fixed, Sticky
- Using Top, Right, Bottom, Left With Positioning
- Z-Index And Stacking Context
- Media Queries For Responsive Designs

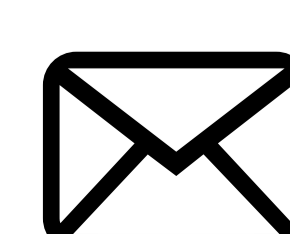
### Project 7: ILUGC Website

- Images: <img>, srcset, alt, picture
- Video: <video>, controls, autoplay, loop, muted
- Audio: <audio>, controls, autoplay, loop, preload
- Embedding YouTube And Other Media
- Grid, Navigation Bar, Display Properties
- Mouse Hover Functionalities

### Project 8: Portfolio Website

- Importance Of Semantic Elements For SEO & Accessibility
- Elements: <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>
- When And Where To Use Semantic Tags
- Creating A Nav Bar, Internal Linking Multiple HTML Pages, Border Radius

**\*\*\*Assignments Will Be Given After Each Module\*\*\***



# JAVASCRIPT

## Module 1: Variables And Date Types

- Understand Var, Let, And Const When And Why To Use Each
- Explore JavaScript's Powerful Dynamic Typing System
- Work With Strings, Numbers, Booleans, Null, Undefined, And More

## Module 2: Operators & Logic

- Arithmetic, Assignment, Comparison, And Logical Operators
- Master The Building Blocks Of Computation And Condition

## Module 3: Conditionals & Control Flow

- Decision-making with if, else, else if
- switch statement
- Ternary Operator
- Write clean, readable branching logic

## Module 4: Loops & Iteration

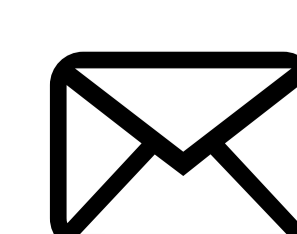
- for, while, and dowhile Loops
- Introduction To forEach() For Working With Arrays

## Module 5: Functions

- Function Declarations
- Function Expressions
- Arrow Functions: Cleaner Syntax, Smarter Scopes
- Parameters, Return Values, And Function Composition
- Call Back Function, Closure, Constructor Function

## Module 6: Objects & Arrays

- Introduction To Object
- Nested Objects, Object Methods
- Create And Manipulate Object Literals
- Understand This In Different Contexts
- Work With Arrays: push, pop, shift, unshift, splice
- Use Powerful Array Methods: map(), filter(), reduce() for real-world data tasks



## JAVASCRIPT

### Module 7: Document Object Model (DOM)

- **Selecting Elements:** Use Document.getElementById, ClassName, tagName, GetElementsByClassName, GetElementsByTagName, QuerySelector & So On
- Change Text, Classes, Styles, Attributes, Creating New Elements, Removing Existing Elements Dynamically
- **Event Handling:** onClick(), onMouseHover(), onDoubleClick(), Add Event Listeners, & So On

### Module 8: Asynchronous

- Synchronous
- Asynchronous
- Callback And Callback Hell
- Promise
- Async/Await
- Application Programming Interface (Fetch and Axios API)

## PROJECTS

### Project 1: Counter App

- Variables, Data Types, Operators
- Interaction With HTML Elements Using DOM

### Project 2: Password Hide/Show

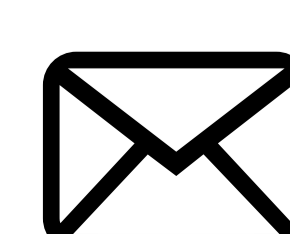
- Inputs Types, Interaction With HTML Attributes Using DOM
- Event Handling (onClick())

### Project 3: RunAwayButton

- Learn Button And Positions, Interaction With CSS Using DOM
- Event Handling (mouseHover())

### Project 4: Typing Speed Checking

- Interaction With HTML Input Box Using DOM
- Event Handling (onInput(), onChange()), Using Arithmetic Operators



## JAVASCRIPT

### Project 5: To-Do List With Local Storage

- Introduction To Local Storage
- Storing And Retrieving Data
- Creating, Updating, And Deleting HTML Elements Dynamically
- Interaction With CSS (text-decoration)

### Project 6: Image Slider

- Functions
- Conditional Statements, Looping
- CSS Positioning
- Event Handling, Basic Animations

### Project 7: Calculator

- Use Inbuilt Functions
- Creating Attractive Calculator UI
- Operators, Conditional Statements, Mouse Hover Animations

### Project 8: Form Validation

- Conditional Statements, Form Attributes, Event Handling
- Interaction With CSS Using DOM
- Validating Inputs Using Conditional Statements

### Project 9: Weather Application

- Fetching Weather Data Using API
- Learn Asynchronous Functions
- Getting Data From JSON as Key And Value
- Using Array Methods: map, forEach
- Sending HTTP Requests

### Project 10: Product Card

- Fetching Product Data Using API, Learn Asynchronous Functions
- Adding Features Like Add To Cart Button
- Looping Through Array Methods
- Event Handling (onClick(), onMouseHover())

\*\*\*Assignments Will Be Given After Each Module\*\*\*



## REACT JS

### Module 1: React Fundamentals

- What Is React? Why React?
- SPA Vs MPA
- Setup Using Vite Or Create React App
- JSX Syntax And Rules
- Functional Components
- Props And Component Reusability
- Conditional Rendering
- Lists And Keys

### Module 2: State Management With Hooks

- useState, useEffect, useRef, useContext
- State Lifting
- Form Handling With Controlled Components

*Projects: Counter Application, Password Hide/Show, Typing Speed Check*

### Module 3: React Router (V6+)

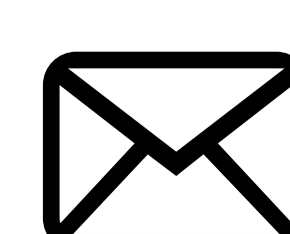
- BrowserRouter, Routes, Route
- Link And NavLink
- Nested Routing
- Dynamic Routing With URL Params
- UseNavigate And Programmatic Routing

*Projects: Multi Page Web App, Toggle Theme*

### Module 4: Working With APIs

- HTTP Methods And REST Principles
- Using Fetch And Axios, API Calls With useEffect
- Handling Loading, Error, And Empty States
- Search And Filtering From API Data

*Projects: Weather App, E-Commerce Web App*



## REACT JS

### Module 5: Advanced Hooks

- UseReducer For Complex State Management
- UseMemo And UseCallback For Optimization
- Custom Hooks
- Debouncing And Throttling With Hooks

### *Projects: Form Validation*

### Module 6: Global State Management

- Understanding Prop Drilling
- Context API With Custom Hooks
- Redux Toolkit: Slices, Store, UseDispatch/UseSelector, Async Thunks
- React Query / TanStack Query (Bonus)

### *Projects: Quiz Application*

### Module 7: Styling In React

- CSS Modules
- Tailwind CSS
- Styled-Components
- Responsive Design Strategies

### Module 8: Code Optimization & Best Practices

- Component Reusability, React.Memo And Pure Components
- Lazy Loading And Code Splitting Using React
- Lazy And Suspense
- Error Boundaries, Folder Structure And Naming Conventions

### Module 10: Deployment & DevOps Basics

- Building React Apps For Production
- Hosting On Netlify, Vercel, Or GitHub Pages
- Using Environment Variables (.Env)
- Introduction To CI/CD Using GitHub Actions
- Above Project Will Be Hosted Using Vercel



# NODE JS

## Modules 1 - 4: Node JS Core Concepts

### Topics:

- What Is Node.Js — V8 Engine, Event Loop, Non-Blocking I/O
- Global Objects: Process, \_\_dirname, \_\_filename
- CommonJS Modules: Require() And Module.Exports
- ES Modules In Node: Import/Export Syntax
- Built-In Modules: Fs (ReadFile, WriteFile, AppendFile), Path, Os
- Npm Init, Package.Json, DevDependencies, Scripts, Nodemon

### Class Activities:

- Run First Node.Js Script From Terminal
- Read/Write/Delete Files Using The Fs Module
- Build A Custom Helper Module And Require It In Another File

### Week Project:

- CLI Notes Manager-Add, List, Read, Delete Notes
- Stored As JSON Via Process.Argv Arguments

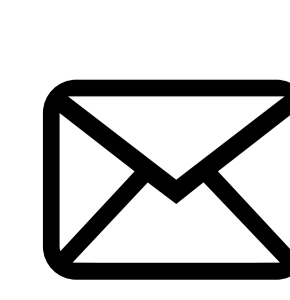
## Modules 5 - 7: HTTP Module, Streams & Environment

### Topics:

- Creating An HTTP Server With The Built-In HTTP Module
- Req And Res Objects - Routing Manually By URL And Method
- Serving HTML, JSON Responses; Status Codes & Headers
- EventEmitter Class - Emit() And On() Pattern
- Readable & Writable Streams; Piping Streams
- Dotenv Package - Environment Variables For Dev Vs Production

### Class Activities:

- Build A Manual File Server Serving Static Assets
- Custom Event Logger Using EventEmitter
- Stream A Large File To HTTP Response Using Pipe

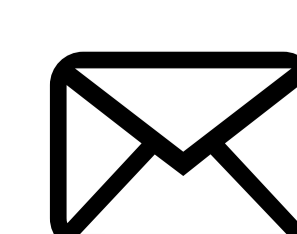


# NODE JS

## Week Project:

- Projects 2 & 3: Static File Server
- HTML/CSS/Images, 404 Handling
- Event-Driven App Logger Writing Timestamped Logs To File

**\*\*\*Assignments Will Be Given After Each Module\*\*\***



# EXPRESS JS

## Modules 1 - 4: Express Routing & Middleware

### Topics:

- Installing Express, Project Folder Structure (MVC)
- App.Listen(); Res.Send(), Res.Json(), Res.Status()
- Route Parameters (Req.Params), Query Strings (Req.Query)
- Request Body Parsing: Req.Body With Express.Json()
- Express.Router() For Modular, Grouped Routes
- Middleware Chain: Custom, Morgan (Logging), Helmet (Security)
- Error-Handling Middleware: 4-Parameter Signature

### Class Activities:

- Build A Basic Express Server And Test In Postman
- Create Modular Routes With Express.Router For Different Resources
- Add Morgan + Custom Request Logger Middleware

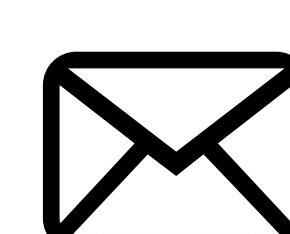
### Week Project:

- API For Notes — Full CRUD
- (GET All, GET By ID, POST, PUT, DELETE)
- With In-Memory Array, Tested In Postman

## Modules 5 - 7: Auth, File Uploads & Error Handling

### Topics:

- Password Hashing With Bcrypt - SaltRounds, Hash, Compare
- JWT: Jsonwebtoken - Sign, Verify, Token Expiry
- Auth Middleware - Protect Routes With Token Verification
- HttpOnly Cookies With Cookie-Parser; CORS Configuration
- Rate Limiting With Express-Rate-Limit
- Multer For Multipart/Form-Data File Uploads; Express.Static()
- Centralized Error Handler; Custom Error Classes; Consistent Error



## EXPRESS JS

### Class Activities:

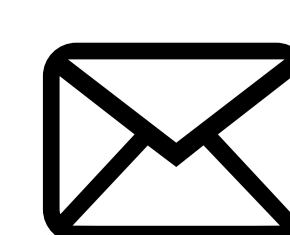
- Build Register/Login Endpoints With Bcrypt + JWT
- Add Auth Middleware To Protect A User Profile Route
- Upload Image File, Save To Disk, Return Public URL

### Week Project:

- User Auth API + Blog API
- With Category Filtering & Pagination + Image
- Upload API With Multer

**\*\*\*Assignments Will Be Given After Each Module\*\*\***

express



# MONGO DB

## Modules 1 - 4: MongoDB & Mongoose Fundamentals

### Topics:

- SQL Vs NoSQL — Key Differences; When To Use Each
- MongoDB Atlas Cloud Setup; MongoDB Compass GUI
- CRUD: InsertOne/Many, Find, FindOne
- UpdateOne/Many, DeleteOne/Many, Upsert
- Query Operators: \$Eq, \$Gt, \$Lt, \$In, \$Nin, \$And, \$Or, \$Regex
- Mongoose: Connect, Schema Definition, Model Creation, Validation
- Model Methods: Save(), Find()
- FindById(), FindByIdAndUpdate(), FindByIdAndDelete()

### Class Activities:

- Set Up MongoDB Atlas Free Cluster And Connect Compass
- Run All CRUD Operations Directly In Compass
- Connect Mongoose To The Express API From Express JS Module

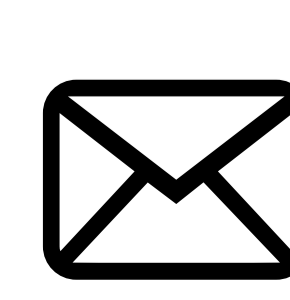
### Week Project:

- Student Records API - Full
- Mongoose CRUD, Filter By Subject Or Grade
- Range Using Query Operators

## Modules 5 - 7: Relationships, Aggregation & Indexes

### Topics:

- Embedding Vs Referencing Documents - When To Use Each
- Populate() For Referenced Relationships Between Collections
- One-To-Many & Many-To-Many Schema Design Patterns
- Mongoose Virtual Fields, Timestamps (CreatedAt, UpdatedAt)
- Aggregation Pipeline: \$Match, \$Group, \$Sort, \$Limit, \$Skip
- \$Lookup (Join Collections), \$Unwind, \$Project, \$Count, \$Sum
- Indexes: Single, Compound, Unique, TTL; Explain() For Performance Analysis



# MONGO DB

## Class Activities:

- Build Users + Posts Schema And Test Populate() In Postman
- Write An Aggregation To Get Total Orders Per Month
- Use Explain() To Compare Indexed Vs Non-Indexed Query Performance

## Week Project:

- Blog API With Populate
- E-Commerce Product Catalog With Search &
- Sort + Analytics Dashboard With Aggregation

**\*\*\*Assignments Will Be Given After Each Module\*\*\***



## INTEGRATION WEEK

### Modules 1 - 4: Full Stack Integration Week

#### Topics:

- Connect React Frontend To Express Backend With Axios
- Axios Base URL Configuration And Request Interceptors
- CORS Setup For Development And Production Environments
- JWT Auth Flow End-To-End
- React Form Fi Express Fi MongoDB Fi JWT Fi React State
- Environment Variables In React (.Env With Vite) And Node
- Protected React Routes Using Stored JWT Token
- File Upload: React Form Fi Multer
- Fi Store Path In MongoDB Fi Return URL To React

#### Class Activities:

- Wire Up React App To Live REST API Endpoints
- Build Complete Login Flow Across All Three Layers
- Upload Image From React Component And Display In UI

#### Week Project:

- Integration Project: Full Stack App With Auth,
- CRUD, And Image Upload - All Four Layers
- Connected And Working Together

### Modules 5 - 7: Capstone Project & Deployment

#### Topics:

- Plan Schema Design And API Routes Before Writing Code
- Feature Scoping What Is Achievable In One Week
- Deploy Frontend To Vercel Or Netlify With Build Settings
- Deploy Backend To Render Or Railway (Free Tier)
- Connect MongoDB Atlas In Production Via Environment Variables
- GitHub Actions CI/CD: Auto-Deploy On Push To Main
- Portfolio Prep: README, Live URL, Clean Code, Demo Video



## INTEGRATION WEEK

### Class Activities:

- Students Pitch Their Capstone Project Idea (Day 1)
- Daily Stand-Ups And Mid-Week Code Review Sessions
- Final Demo Day - Live Presentation To Class

### Week Project:

- Capstone Project: Student-Choice Full
- MERN App (E-Commerce, Job Board, Chat
- App, Or Project Manager): Deployed With Live URL



# “ OUR TRAINEES GOT PLACED IN ”

## Connect With Us!

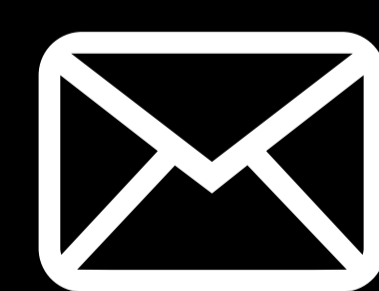


[www.payilagam.com](http://www.payilagam.com)



+91 8344 777 333

+91 8883 775 533



[info@payilagam.com](mailto:info@payilagam.com)



No: 7, 1st Main Rd,  
BHEL Shakthi Nagar,  
Vijaya Nagar,  
Velachery, Chennai,  
Tamil Nadu 600042