WHITE SCHOLARS

CERTIFIED

FULL STACK COURSE

CLASSROOM | ONLINE

IN ASSOCIATION WITH

Microsoft

WHITE SCHOLARS

Why Trust Us?

- 100+ Batches
- Dedicated placement support
- 25+ hiring partners
- Senior Data scientists as faculty
- 2000+ passed out students
- Guaranteed internship opportunity
- Placement assistance

Companies that hire for Full Stack

WHITE SCHOLARS

FAQS?

Duration

Weekday: 4 Months

Weekends: 5 Months

Laptop Configuration

OS: Windows X,

Ram: 6GB, HDD: 1TB

Eligibility

BE/MBA/BSc/B.Tech.MSC

M.Tech - IT/Statistics/CSE

Avg. Packages

3.2-5 LPA (Fresher)

5-15 LPA (Experienced)

Certifications

White Scholars and microsoft

Pre Requisite

No pre-requisites to learn Full Stack

Roles

Backend Developer, Frontend Developer, MERN Stack Developer, Database Admin, Database Manager, Website Developer, Software Developer, Website Designer, Backend Engineer, Node.JS Developer, Express JS Developer, React JS Developer

Training Flow

1. Course Enrollment
2. Live Classes
3. Weekly Assignments
4. Project Work
5. Completion of Modules
6. Job Applications
7. Mock Interviews
8. Resume Improvements
9. Calls from HRs
10. Interviews and Tests
11. Offer Letter

Month on Month Journey

1 Introduction and Setup

- Overview of Web Development
- JavaScript Refresher
- Advanced JavaScript Concepts
- Introduction to Node.js

2 MongoDB and Express.js

- Introduction to NoSQL and MongoDB
- CRUD Operations in MongoDB
- Introduction to Express.js
- RESTful APIs

3 React.js Basics and Advanced Concepts

- Introduction to React.js
- Event Handling and Forms
- React Router
- Context API and Advanced Hooks

4 Integration, Deployment, and Live Project

- Create Live Project
- E-Commerce App Development
- Social Media App
- Single and Multivendor app

5 Apply for Job and Internship

- Identify the right jobs and apply on Naukri | Linkedin
- Analytics Vidhya | Datajobs
- Kaggle Job Portal
- Internshala
- indeed.co

Curriculum

Modules

JavaScript

Node.JS

NoSQL

React JS

Mongo DB

HTML/CSS

Introduction and Setup

WEEK 1-2

Basics

1. Overview of Web Development
- Understanding client-server architecture
- Introduction to front-end and back-end technologies
- Overview of MERN stack (MongoDB, Express.js, React.js, Node.js)

Setting Up Development Environment

- Installing Node.js and npm
- Setting up MongoDB locally and using cloud services (MongoDB Atlas)
- Installing VS Code or other IDEs
- Introduction to version control with Git and GitHub
- Basic Git commands and workflows

JavaScript Refresher

- ES6+ features (let/const, template literals, destructuring, spread/rest operators)
- Functions (arrow functions, callbacks, higher-order functions)
- Asynchronous JavaScript (promises, async/await)
- JavaScript DOM manipulation

WEEK 3-4

Advanced JavaScript Concepts

- Scope, closures, and the 'this' keyword
- Prototypes and inheritance
- Modules and namespaces
- Error handling and debugging

Introduction to Node.js

- Node.js architecture and its asynchronous nature
- Understanding the event loop and event-driven programming
- Core modules (fs, path, http, etc.)
- Building a simple HTTP server

MongoDB and Express.js

WEEK 5-6

Introduction to NoSQL and MongoDB

- Differences between SQL and NoSQL databases
- JSON/BSON and document-based storage
- Setting up MongoDB locally and on the cloud

CRUD Operations in MongoDB

- Creating, reading, updating, and deleting documents
- MongoDB queries and aggregations
- Indexing and performance optimization

Mongoose ODM

- Setting up Mongoose
- Creating schemas and models
- Data validation and schema design
- Middleware and hooks in Mongoose

WEEK 7-8

Introduction to Express.js

- Setting up an Express server
- Understanding middleware and routing
- Handling different types of requests (GET, POST, PUT, DELETE)

Building RESTful APIs

- REST principles and best practices
- Structuring API routes and controllers
- Implementing CRUD operations
- Error handling and logging

Advanced Express.js Concepts

- Middleware functions (built-in and third-party)
- Handling file uploads with Multer
- Authentication and authorization (basic auth, JWT)

React.js Basics and Advanced Concepts

WEEK 9-10

Introduction to React.js

- What is React and why use it?
- Setting up a React environment with Create React App
- JSX syntax and expressions
- Functional components and props

State and Lifecycle in React

- Understanding state and setState
- Component lifecycle methods
- Introduction to Hooks (useState, useEffect)

Event Handling and Forms

- Handling events in React
- Controlled vs uncontrolled components
- Building and validating forms

WEEK 11-12

React Router

- Setting up React Router
- Defining routes and nested routes
- Programmatic navigation

State Management with Redux

- Introduction to Redux and its core principles
- Setting up Redux in a React project
- Actions, reducers, and the Redux store
- Connecting React components to Redux

Context API and Advanced Hooks

- Using the Context API for state management
- Custom hooks and reusable logic
- Optimizing performance with React.memo and useCallback

Integration, Deployment, and Live Project

WEEK 13-14

Making HTTP Requests

- Using fetch API and Axios for AJAX calls
- Handling API responses and errors
- CORS and cross-origin requests

Authentication and Authorization

- Implementing JWT-based authentication
- Protecting routes and user roles
- Securing API endpoints with middleware

WEEK 15-16

E-commerce App Development

- Defining project requirements and scope
- Implementing features: user authentication, product listings, shopping cart, checkout process
- Integrating front-end and back-end
- Testing and debugging

Final Project Presentation

- Presenting the e-commerce app
- Code review and feedback
- Future learning paths and resources

WEEK 19-20

Live Project Deployment

- Defining project requirements and scope
- Implementing features and functionalities
- Integrating front-end and back-end
- Testing and debugging

Final Project Presentation

- Presenting the live project
- Code review and feedback
- Future learning paths and resources

Prerequisites:

- No prerequisite required. Anyone can learn.

Recommended Tools:

- Visual Studio Code
- Git and GitHub
- Postman for API testing
- MongoDB Compass

Learning Outcomes:

- Develop and deploy full-stack applications using the MERN stack.
- Understand and implement modern web development practices.
- Gain proficiency in both front-end and back-end development.
- Build a portfolio of projects showcasing your skills.