XCODERS
IT Training & Placements

100% PLACEMENT ASSISTANCE

MERN STACK

(MongoDB, Express.js, React.js, & Node.js)

💥 **Why Choose XCoders?**

🛠️ Hands-on projects for real-world experience

🌟 Guaranteed Internships with stipends up to ₹10,000!

💬 (Terms and conditions apply)

🎯 100% Placement Assistance to help you secure your dream job!

📄 5 to 10 Interview Calls Tailored support to help you land your ideal job.

Module 1: Web Development Fundamentals

Students will have a solid understanding of how the web works, the structure of web pages, and styling using HTML and CSS. They will be able to create responsive websites and interact with web elements using JavaScript.

Introduction to Web Development

HTML5:

- Document structure, semantic elements **(<header>, <footer>, <section>, etc.)**
- Forms and form validation
- Media elements **(<audio>, <video>, <canvas>)**

CSS3:

- Box Model (margin, padding, borders)
- Layout **(Flexbox, Grid)**
- Transitions and Animations
- Media Queries for Responsive Design

JavaScript Basics:

- Variables **(let, const, var)**
- Data Types **(String, Number, Boolean, Object, Array)**
- Control Flow **(if-else, switch, loops)**
- Functions **(Declaration, Expression, Arrow Functions)**

Course Project

Portfolio Website

Build a Responsive Portfolio Website that showcases personal projects, has a contact form, and adapts to different screen sizes.

Advanced JavaScript Concepts

Objects & Arrays:

- Creating and manipulating arrays and objects
- Array methods (map(), filter(), reduce(), forEach())
- Object-oriented programming in JavaScript (Classes, Prototypes)

DOM Manipulation:

- Accessing and modifying HTML elements (document.getElementById(), querySelector())
- Adding and removing elements dynamically
- Event handling (addEventListener(), mouse and keyboard events)

ES6+ Features:

- Template Literals, Destructuring, Spread & Rest Operators
- Promises, async/await for handling asynchronous operations

Course Project

To-Do List Application

Develop an interactive To-Do List Application where users can add, edit, and delete tasks using JavaScript. Focus on DOM manipulation and event handling.

Git & Version Control

Git Basics:

- Installing Git and initial setup
- Git workflow: git init, git add, git commit, git push
- Branching and merging (git checkout, git merge)

GitHub:

- Creating and managing repositories
- Collaborating on projects (Pull Requests, Issues)
- GitHub Pages for project hosting
- Creating and manipulating arrays and objects
- Array methods (map(), filter(), reduce(), forEach())
- Object-oriented programming in JavaScript (Classes, Prototypes)s

Course Project

To-Do List Application

Manage the version control of your To-Do List App using Git and GitHub. Collaborate with a partner on GitHub for code reviews and pull requests.

Module 2: Frontend Development with React.js

students will be comfortable with building component-based UIs, managing state, and rendering dynamic content. They will have built a scalable React application with real-world features.

Introduction to React.js

React Basics:

- Introduction to JSX **(JavaScript XML),** rendering elements
- Functional and Class Components
- Props and State: passing data between components, managing state within components
- Component lifecycle methods in class components **(componentDidMount(), componentDidUpdate())**

Event Handling:

- Handling events **(click, change, submit)**
- Binding event handlers
- Conditional rendering based on state

React.js Advanced Features

React Router:

- Setting up React Router for navigation between pages
- Route parameters and navigation hooks **(useNavigate(), useParams())**
- Nested Routes, dynamic routes

React Hooks:

- Introduction to hooks: **useState(), useEffect()**
- Handling side effects with useEffect()
- Managing global state with useContext()

Context API:

- Creating context, sharing state globally between components
- Provider and Consumer pattern
- Replacing Redux with Context API for state management

📁 **Course Project**

Personal Blog Application

Enhance the Personal Blog Application by adding React Router for navigation between pages (Home, Blog List, Single Post View) and implement Context API for global state management.

Module 3: Backend Development with Node.js and Express.js

Students will be able to build server-side applications using Node.js, handling asynchronous operations, and creating APIs for frontend applications to consume.

Introduction to Node.js

Node.js Basics:

- Introduction to Node.js and event-driven architecture
- Using Node Package Manager **(NPM)** to install packages
- Creating a simple HTTP server

Asynchronous JavaScript:

- Callbacks vs. Promises
- Introduction to async/await for handling asynchronous operations cleanly
- File system operations **(Reading/Writing files using fs module)**

Express.js for API Development

Express.js Basics:

- Setting up Express, routing, and middleware
- Handling HTTP methods: GET, POST, PUT, DELETE
- Query parameters, route parameters

Middleware:

- Built-in middleware (body-parser, static files)
- Custom middleware for logging, authentication

Error Handling:

- Try-catch blocks for handling asynchronous errors
- Centralized error handling middleware

User Authentication:

- JSON Web Tokens (JWT) for secure authentication
- Creating a login and registration system
- Protecting routes with authentication middleware

Course Project
User Authentication System

Build a Simple REST API to manage users (CRUD operations) in Node.js using basic server-side concepts and file system interactions. Expand the REST API by creating a User Authentication System. Implement JWT-based authentication and secure API endpoints.

Database Integration with MongoDB

Introduction to MongoDB:
- MongoDB overview, collections, and documents
- CRUD operations with MongoDB (Insert, Find, Update, Delete)
- Using MongoDB Atlas for cloud databases

Mongoose for Data Modeling:
- Defining schemas and models
- Data validation, pre and post hooks
- Relationships between collections (One-to-Many, Many-to-Many)

Module 4: Backend Development with Node.js and Express.js

Students will be able to build server-side applications using Node.js, handling asynchronous operations, and creating APIs for frontend applications to consume.

Full-Stack Project (Frontend + Backend Integration)

Connecting React with Node/Express:
- Fetching data from Node/Express APIs using axios or fetch()
- Form submissions in React and sending POST requests to Express
- Handling authentication (JWT tokens) on the frontend

Frontend-Backend Communication:
- Using state to reflect backend changes in the UI (Add, Update, Delete)
- Error handling (displaying validation errors from the backend)

Authentication and Authorization

User Authentication:
- Implementing secure login, registration, and password hashing
- Using JWT tokens for authorization in the backend
- Protecting private routes on the frontend with React Router

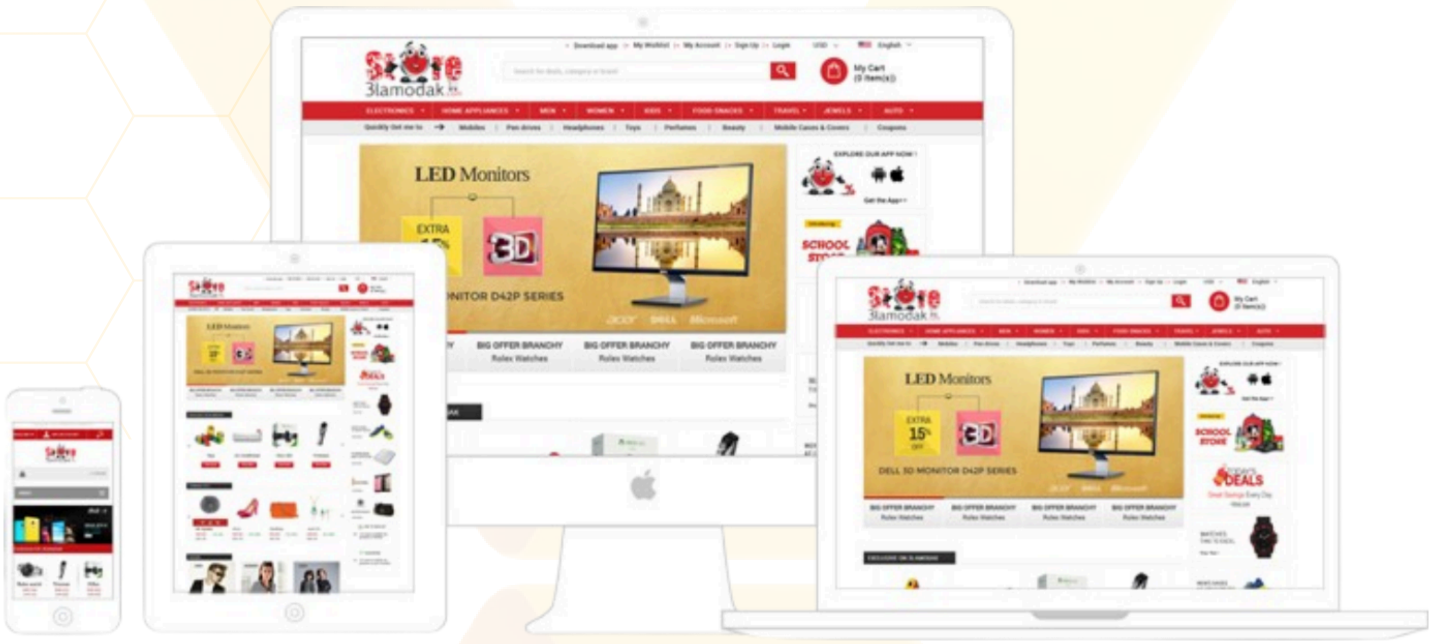
Role-Based Access Control:
- Implementing roles **(Admin, User)** in MongoDB
- Securing admin routes and features

Deployment:
- Deploying full-stack applications
- Environment variables and production configuration
- Optimizing performance (lazy loading, code splitting)

Final Project

E-commerce Website

Deploy a fully functional
Full-Stack E-commerce Website like: Flipkart, Amazon, Ajio and croma.



Final Project

OTT Movies Platform

Deploy a fully functional
Full-Stack OTT Platform like: Netflix

NETFLIX