

# Full Stack Mobile Application Development with React Native and Node

**Course Duration: 80 Hours**

**Course code: FSMADRNN**

## 1. Course Overview

This course provides an end-to-end learning path for building, managing, and deploying full-stack mobile applications using React Native for cross-platform (iOS & Android) mobile development and Node.js with Express.js for backend APIs. Learners will understand mobile UI/UX design, API integration, authentication, database management, real-time features, and cloud deployment. By the end of the course, participants will be capable of designing, developing, and deploying production-ready mobile applications with a Node.js-powered backend.

## 2. What you'll learn?

**By the end of this course, learners will be able to:**

- Understand mobile application development architecture.
- Build cross-platform mobile apps with React Native.
- Create RESTful APIs using Node.js and Express.js.
- Integrate mobile apps with databases (SQL & NoSQL).
- Implement authentication & authorization with JWT/OAuth.
- Work with native modules, device APIs, and push notifications.
- Manage state using Redux/Context API.
- Handle file uploads, media storage, and cloud integration.
- Test, debug, and optimize mobile applications.
- Deploy mobile apps to App Store, Google Play, and backend on cloud.

### 3. Target Audience

- Developers aiming to become full-stack mobile developers.
- React or JavaScript developers wanting to build mobile-first apps.
- Teams or individuals looking to build scalable cross-platform mobile apps.
- Professionals preparing for mobile app development careers.

### 4. Pre-Requisites

#### **Familiarity with:**

- JavaScript/ES6 concepts.
- Basic understanding of React.js.
- Basic programming and database knowledge.

### 5. Course content

#### **Day 1: Introduction to Node.js & Express – REST API Fundamentals**

##### **Theory (2.5 hrs)**

- What is Node.js? Use cases in mobile backend
- Node.js architecture (event loop, async I/O)
- Introduction to Express.js
- RESTful concepts (resources, verbs, status codes)
- Project structure and best practices

##### **Hands-on (5.5 hrs)**

- Initialize project, install Express
- Create REST API with GET, POST routes
- Use nodemon for hot-reloading
- Test APIs using Postman
- Organize routes with Express Router

## **Day 2: Express Routing, Middleware & Stateless Authentication**

### **Theory (3.0 hrs)**

- Express Router in-depth
- Custom middleware functions
- HTTP lifecycle in Express
- Introduction to input validation

### **Hands-on (5.0 hrs)**

- Modularize /users, /auth routes
- Add custom middleware (logger, validator)
- Handle JSON and form data
- Create simple login/register flow (no DB yet)

## **Day 3: MongoDB Integration & JWT Authentication**

### **Theory (3.0 hrs)**

- NoSQL basics and MongoDB Atlas overview
- Introduction to Mongoose (schema, model)
- JWT authentication, hashing with bcrypt
- CORS, Helmet, and security best practices

### **Hands-on (5.0 hrs)**

#### **Connect to MongoDB**

- Create User schema with Mongoose
- Build full login/register API with hashed passwords
- Implement JWT token generation and verification
- Secure private routes using token middleware

## **Day 4: Mobile API Features & Backend Deployment**

### **Theory (2.5 hrs)**

- Structuring scalable APIs for mobile

- Filtering, pagination, and sorting
- Deployment platforms: Render, Railway, Vercel
- Managing environment variables

### **Hands-on (5.5 hrs)**

- Add filtered /tasks endpoint with pagination
- Use .env for secrets
- Deploy backend using Render with GitHub integration
- Final backend mini project: secure CRUD task manager

## **Day 5: Firebase for Cloud Integration**

### **Theory (3.0 hrs)**

- Firebase overview (Auth, Firestore, Storage, Functions)
- Firebase vs traditional backend
- Emulator suite, pricing, and best use cases

### **Hands-on (5.0 hrs)**

- Initialize Firebase project
- Implement Auth with email/password
- CRUD in Firestore with snapshot listeners
- Upload/download files with Firebase Storage
- Deploy a simple Cloud Function (Node.js)

## **Day 6: React Native – Setup & Fundamentals**

### **Theory (3.0 hrs)**

- Introduction to React Native (Expo & CLI)
- JSX, components, styles
- Flexbox layout, ScrollView, SafeAreaView

### **Hands-on (5.0 hrs)**

- Setup development environment (Expo or CLI)
- Build a simple mobile UI with View, Text, Image
- Implement layouts using Flexbox and ScrollView
- Navigation between static screens using Stack Navigator

## **Day 7: Navigation & State Management in React Native**

### **Theory (2.5 hrs)**

- React Navigation: stack, tab, drawer
- State: useState, useEffect, Context API
- Props vs state, lifting state up

### **Hands-on (5.5 hrs)**

- Implement a login flow with navigation guard
- Use Context API for auth and global state
- Build reusable UI components
- Persist login session with AsyncStorage

## **Day 8: Performance Optimization & Memory Management**

### **Theory (2.5 hrs)**

- Optimizing FlatList, avoiding re-renders
- Memoization: useMemo, useCallback
- Flipper, Hermes, and profiling tools

### **Hands-on (5.5 hrs)**

- Optimize rendering of dynamic lists
- Implement FastImage for caching
- Profile app memory and CPU
- Compare Hermes-enabled and default builds

## **Day 9: Animations, Gestures & Device Feature Access**

### **Theory (2.5 hrs)**

- Reanimated 2 & Gesture Handler overview
- Accessing native APIs: Camera, GPS, Biometrics, Sensors
- Permission handling (iOS/Android)

### **Hands-on (5.5 hrs)**

- Create animated swipe card UI
- Use react-native-geolocation and maps
- Implement fingerprint login with keychain
- Use image-picker and sensors for motion tracking
- Setup background task using background-fetch

## **Day 10: Real-time Features, Push Notifications & CI/CD**

### **Theory (3.0 hrs)**

- Real-time options: WebSocket vs Firebase
- Local & Remote Notifications with FCM
- Secure local storage (tokens, credentials)
- CI/CD: GitHub Actions, CodePush, Fastlane

### **Hands-on (5.0 hrs)**

- Create real-time chat using Firebase or Socket.IO
- Send push notifications from Firebase
- Secure sensitive data using encrypted-storage
- Set up OTA updates and auto builds with GitHub CI