

# MERN Stack Internship – Daily Task Sheets (4 Months)

## Daily Format (Recommended)

- 📖 Learning (Theory)
  - 📝 Practice Task
  - ☐ Assignment
  - ✨ Output Expected
- 

## MONTH 1 – JavaScript, TypeScript, HTML, CSS & Web Basics

---

### Week 1 – JavaScript, TypeScript & Git

#### Day 1 – Orientation & Setup

- **Learning:**
    - Internship overview
    - MERN + TypeScript roadmap
    - Expectations & coding standards
  - **Practice:**
    - Install Node.js
    - Install VS Code
    - Install Git
  - **Assignment:**
    - Create GitHub account
  - **Output:**
    - GitHub profile link
- 

#### Day 2 – JavaScript Fundamentals

- **Learning:**
  - Variables, data types
  - `let`, `const`
- **Practice:**
  - Write basic JS programs
- **Assignment:**

- Even / Odd number program
  - **Output:**
    - Working JS file committed to GitHub
- 

## Day 3 – Functions & Arrays (JavaScript)

- **Learning:**
    - Functions
    - Arrays & array methods
  - **Practice:**
    - Array operations
  - **Assignment:**
    - Find max & min from an array
  - **Output:**
    - JS logic with clean functions
- 

## Day 4 – Introduction to TypeScript

- **Learning:**
    - What is TypeScript & why we use it
    - TypeScript vs JavaScript
    - tsconfig.json
  - **Practice:**
    - Setup TypeScript project
  - **Assignment:**
    - Write first TypeScript program
  - **Output:**
    - Running TypeScript file (.ts)
- 

## Day 5 – Git & GitHub Workflow

- **Learning:**
    - Git clone, commit, push
    - Basic branching
  - **Practice:**
    - Create repository
  - **Assignment:**
    - Push JS & TS code to GitHub
  - **Output:**
    - GitHub repo link
-

# Week 2 – TypeScript + HTML + CSS

## Day 6 – TypeScript Types & Interfaces

- **Learning:**
    - Primitive types
    - Interfaces & type aliases
  - **Practice:**
    - Create typed objects
  - **Assignment:**
    - Define `User` interface
  - **Output:**
    - Strongly typed TS file
- 

## Day 7 – TypeScript Functions & Enums

- **Learning:**
    - Typed functions
    - Enums & union types
  - **Practice:**
    - Utility functions
  - **Assignment:**
    - Create reusable typed functions
  - **Output:**
    - Clean TypeScript utilities
- 

## Day 8 – HTML Basics

- **Learning:**
    - HTML structure
    - Semantic tags
  - **Practice:**
    - Create basic webpage
  - **Assignment:**
    - Profile page
  - **Output:**
    - HTML file
- 

## Day 9 – CSS & Responsive Design

- **Learning:**
  - Flexbox
  - Media queries

- **Practice:**
    - Responsive layout
  - **Assignment:**
    - Mobile-friendly page
  - **Output:**
    - Responsive UI
- 

## Day 10 – Mini Project (HTML + CSS + TS)

- **Project:**
    - Static Login Page
  - **Assignment:**
    - Add TypeScript form validation
  - **Output:**
    - Login UI with TS validation
- 

## Week 3 – Advanced TypeScript & Browser APIs

### Day 11 – DOM Manipulation with TypeScript

- **Learning:**
    - DOM types
    - Type-safe selectors
  - **Practice:**
    - DOM updates
  - **Assignment:**
    - Dynamic text change
  - **Output:**
    - Typed DOM application
- 

### Day 12 – Events & Forms (TypeScript)

- **Learning:**
    - Event types
    - Form handling
  - **Practice:**
    - Form submit handling
  - **Assignment:**
    - Client-side validation
  - **Output:**
    - Validated form using TS
-

## Day 13 – Async Programming

- **Learning:**
    - Promises
    - `async` / `await`
  - **Practice:**
    - Fetch API
  - **Assignment:**
    - Fetch and display user list
  - **Output:**
    - Typed API response handling
- 

## Day 14 – Local Storage with TypeScript

- **Learning:**
    - `localStorage`
    - Typed data persistence
  - **Practice:**
    - Save & retrieve data
  - **Assignment:**
    - Persist to-do list
  - **Output:**
    - Data retained on refresh
- 

## Day 15 – TypeScript Mini Project

- **Project:**
    - To-Do App using TypeScript
  - **Output:**
    - Fully functional TS app
- 

## Week 4 – Frontend Mini Project

### Day 16 – Project Setup

- **Learning:**
    - Project structure
  - **Practice:**
    - Setup dashboard UI
  - **Output:**
    - Base UI layout
-

## Day 17 – CRUD Operations

- **Practice:**
    - Create, Read, Update, Delete items
  - **Output:**
    - Typed CRUD functionality
- 

## Day 18 – Validation & Error Handling

- **Learning:**
    - Error handling best practices
  - **Output:**
    - Stable UI
- 

## Day 19 – Refactor to Strict TypeScript

- **Practice:**
    - Convert loose JS to strict TS
  - **Output:**
    - Clean & safe codebase
- 

## Day 20 – Review & Demo

- **Activity:**
    - Project demo
    - Code review
  - **Output:**
    - Reviewed & approved frontend project
- 

# MONTH 2 – React + TypeScript (Frontend Development)

---

## Week 5 – React + TypeScript Fundamentals

### Day 21 – React + TypeScript Setup

- **Learning:**

- Why React with TypeScript
    - Vite + React + TS setup
    - Folder structure
  - **Practice:**
    - Create React TS project
  - **Assignment:**
    - Run first React TS app
  - **Output:**
    - React app running on localhost
- 

## Day 22 – Components, Props & State

- **Learning:**
    - Functional components
    - Typed props
    - `useState` with types
  - **Practice:**
    - Create reusable components
  - **Assignment:**
    - Counter app with TypeScript
  - **Output:**
    - Typed React components
- 

## Day 23 – `useEffect` & API Integration

- **Learning:**
    - `useEffect`
    - API calling with types
  - **Practice:**
    - Fetch data from API
  - **Assignment:**
    - Display user list
  - **Output:**
    - Typed API response rendering
- 

## Day 24 – React Router

- **Learning:**
  - React Router setup
  - Typed route params
- **Practice:**
  - Multi-page navigation
- **Assignment:**
  - Dashboard & Profile routes

- **Output:**
    - Working routing system
- 

## Day 25 – Forms & Validation

- **Learning:**
    - Controlled components
    - Typed form handling
  - **Practice:**
    - Login & Signup forms
  - **Assignment:**
    - Client-side validation
  - **Output:**
    - Validated forms in React TS
- 

## Week 6 – State Management & UI Patterns

### Day 26 – Context API

- **Learning:**
    - Context API with TypeScript
  - **Practice:**
    - Global auth state
  - **Assignment:**
    - Auth context provider
  - **Output:**
    - Typed global state
- 

### Day 27 – Error Handling

- **Learning:**
    - Error boundaries
  - **Practice:**
    - Fallback UI
  - **Assignment:**
    - Global error handler
  - **Output:**
    - Stable React application
- 

## Day 28 – Tables & Pagination



- **Learning:**
    - Table components
    - Pagination logic
  - **Practice:**
    - Paginated table
  - **Assignment:**
    - User list with pagination
  - **Output:**
    - Paginated data display
- 

## **Day 29 – Filters & Search**

- **Learning:**
    - Search & filter patterns
  - **Practice:**
    - Client-side filtering
  - **Assignment:**
    - Searchable table
  - **Output:**
    - Filtered UI results
- 

## **Day 30 – React Mini Project (Admin Panel – Phase 1)**

- **Project:**
    - Admin Panel UI
  - **Output:**
    - Base Admin UI with routing
- 

# **Week 7 – Advanced React + TypeScript**

## **Day 31 – Custom Hooks**

- **Learning:**
    - Writing custom hooks with TS
  - **Practice:**
    - Reusable data hook
  - **Assignment:**
    - `useFetch` hook
  - **Output:**
    - Clean reusable hook
-

## Day 32 – Performance Optimization

- **Learning:**
    - `useMemo`, `useCallback`
  - **Practice:**
    - Optimize re-renders
  - **Assignment:**
    - Performance tuning
  - **Output:**
    - Optimized React app
- 

## Day 33 – Advanced Forms

- **Learning:**
    - Complex forms
    - Validation patterns
  - **Practice:**
    - Multi-step form
  - **Assignment:**
    - Form with validations
  - **Output:**
    - Robust form handling
- 

## Day 34 – Code Refactoring

- **Learning:**
    - Clean code principles
  - **Practice:**
    - Convert loose TS to strict
  - **Output:**
    - Clean, strict TypeScript code
- 

## Day 35 – Admin Panel (Phase 2)

- **Project:**
    - Complete Admin Panel UI
  - **Output:**
    - Feature-complete frontend
- 

## Week 8 – Testing, Deployment & Review

## Day 36 – Testing Basics

- **Learning:**
    - React testing fundamentals
  - **Practice:**
    - Component tests
  - **Output:**
    - Basic test cases
- 

## Day 37 – Documentation

- **Learning:**
    - Writing README
  - **Practice:**
    - Project documentation
  - **Output:**
    - Well-documented frontend
- 

## Day 38 – Frontend Deployment

- **Learning:**
    - Vercel deployment
  - **Practice:**
    - Deploy Admin Panel
  - **Output:**
    - Live frontend URL
- 

## Day 39 – Review & Assessment

- **Activity:**
    - Demo
    - Code review
  - **Output:**
    - Reviewed frontend project
- 

## Day 40 – UI/UX Improvements

- **Activity:**
  - Polish UI
- **Output:**
  - Improved user experience

---

# MONTH 3 – Backend (Node.js + Express + MongoDB + TypeScript)

---

## Week 9 – Backend Foundations

### Day 41 – Node.js + TypeScript Setup

- **Learning:**
  - Node + TS architecture
- **Practice:**
  - Setup backend project
- **Output:**
  - Running TS server

---

### Day 42 – Project Structure

- **Learning:**
  - MVC / Service-layer pattern
- **Practice:**
  - Folder structure
- **Output:**
  - Clean backend structure

---

### Day 43 – Express Routes & Controllers

- **Learning:**
  - Typed controllers
- **Practice:**
  - Create CRUD APIs
- **Output:**
  - Working APIs

---

### Day 44 – Middleware

- **Learning:**
  - Middleware concepts
- **Practice:**

- Logger middleware
  - **Output:**
    - Logged API requests
- 

## **Day 45 – MongoDB & Mongoose**

- **Learning:**
    - Schemas & models
  - **Practice:**
    - Typed models
  - **Output:**
    - Connected database
- 

## **Week 10 – Authentication & Business Logic**

### **Day 46 – Database Relationships**

- **Learning:**
    - References & relations
  - **Output:**
    - Linked collections
- 

### **Day 47 – Service Layer**

- **Learning:**
    - Business logic separation
  - **Output:**
    - Maintainable backend
- 

### **Day 48 – Authentication (JWT)**

- **Learning:**
    - JWT flow
  - **Output:**
    - Secure login API
- 

### **Day 49 – Role-Based Access**

- **Learning:**
    - RBAC
  - **Output:**
    - Role-protected APIs
- 

## **Day 50 – Backend Mini Project**

- **Project:**
    - Secure CRUD backend
  - **Output:**
    - Feature-complete backend
- 

## **Week 11 – MERN Integration**

### **Day 51 – Frontend + Backend Integration**

- **Output:**
    - Full-stack auth flow
- 

### **Day 52 – CORS & Config**

- **Output:**
    - Stable integration
- 

### **Day 53 – Pagination**

- **Output:**
    - Scalable APIs
- 

### **Day 54 – Search**

- **Output:**
    - Searchable APIs
- 

### **Day 55 – Validation**

- **Learning:**
    - Zod / Joi
  - **Output:**
    - Safe input handling
- 

## **Week 12 – Stability & Performance**

### **Day 56 – File Uploads**

- **Output:**
    - Image upload API
- 

### **Day 57 – Logging**

- **Output:**
    - Winston logging
- 

### **Day 58 – Performance**

- **Output:**
    - Indexed queries
- 

### **Day 59 – Security**

- **Output:**
    - Hardened backend
- 

### **Day 60 – Review**

- **Output:**
  - Stable backend system

---

# MONTH 4 – Final Full-Stack TypeScript Project & Deployment

---

## Week 13 – Project Design & Build

### Day 61 – Project Planning

- **Output:**
    - ERD & architecture
- 

### Day 62 – Backend Development

- **Output:**
    - CRUD APIs
- 

### Day 63 – Frontend Development

- **Output:**
    - UI screens
- 

### Day 64 – Authentication Integration

- **Output:**
    - Secure full-stack app
- 

### Day 65 – End-to-End Integration

- **Output:**
    - Working application
- 

## Week 14 – Features & Deployment



## Day 66 – Filters & Search

- **Output:**
    - Enhanced UX
- 

## Day 67 – Reports

- **Output:**
    - CSV exports
- 

## Day 68 – Backend Deployment

- **Output:**
    - Live APIs
- 

## Day 69 – Frontend Deployment

- **Output:**
    - Live application
- 

## Day 70 – Testing

- **Output:**
    - Stable production app
- 

## Week 15 – Hardening & Review

### Day 71 – Documentation

- **Output:**
    - API docs (Swagger)
- 

### Day 72 – Code Review

- **Output:**
  - Clean codebase

---

## **Day 73 – Security Audit**

- **Output:**
  - Secure system

---

## **Day 74 – Performance Optimization**

- **Output:**
  - Optimized app

---

## **Day 75 – Demo Preparation**

- **Output:**
  - Demo-ready project

---

# **Week 16 – Closure**

## **Day 76 – Final Demo**

- **Output:**
  - Project presentation

---

## **Day 77 – Technical Evaluation**

- **Output:**
  - Assessment scores

---

## **Day 78 – Feedback & Improvements**

- **Output:**
  - Final enhancements

---

## **Day 79 – Knowledge Transfer**

- **Output:**
    - Documentation & handover
- 

## **Day 80 – Internship Closure**

- **Output:**
  - Final report & certificate