## **CREATING A NEW PROJECT**

### **2.1 Why Version Locking Matters**

React Native CLI projects often break because:

* RN core updates faster than libraries
* Navigation, native modules, and SDKs have strict version ranges
* Android/iOS build tools change over time

If you install “latest everything,” you might get mismatched versions that compile today but fail tomorrow.  
So we’ll use a known working RN version from the official compatibility table.

### **2.2 Check Official Version Compatibility**

Official table: <https://reactnative.dev/versions>  
Example:  
If the latest stable is 0.75.4, use that exact version for safety.

### **2.3 Create Project with TypeScript Template**

npx react-native@latest init MyApp \

--version 0.75.4 \

--template react-native-template-typescript

Why TypeScript template?

* Strong typing avoids runtime bugs
* Better autocompletion in VS Code
* More maintainable large projects

### **2.4 Verify the Project Runs**

cd MyApp

yarn install

npx react-native run-android # For Android

npx react-native run-ios # For iOS (macOS only)

### **2.5 Recommended Folder Structure for Scalability**

MyApp/

src/

assets/ # fonts, images, icons

components/ # reusable UI components

navigation/ # stack, tab, drawer navigators

screens/ # screen components

services/ # API calls, backend services

hooks/ # custom hooks

utils/ # helper functions

constants/ # colors, typography, config

types/ # TypeScript types/interfaces

App.tsx # entry point

react-native.config.js # asset linking config

### **2.6 Asset Linking (Fonts/Images)**

For fonts/images to work properly:

1. Create a react-native.config.js file in CommonJS format:

// react-native.config.js

module.exports = {

assets: ['./src/assets/fonts'],

};

1. Link assets:

npx react-native-asset

### **2.7 Initial Cleanup**

* Remove unused boilerplate files from /src
* Replace App.tsx with a minimal starter:

import React from 'react';

import {SafeAreaView, Text, StyleSheet} from 'react-native';

const App = () => {

return (

<SafeAreaView style={styles.container}>

<Text>Hello React Native CLI 🚀</Text>

</SafeAreaView>

);

};

const styles = StyleSheet.create({

container: {flex: 1, justifyContent: 'center', alignItems: 'center'},

});

export default App;

✅ At the end, you have:

* A fresh RN CLI project with TypeScript
* Version-safe RN installation
* Working Android/iOS build
* Scalable folder structure
* Asset linking ready for fonts/images