**RESEARCH**

**1.** Unit testing is a critical aspect of software development that ensures individual components or functions of an application behave as expected. With the growing complexity of modern applications, the use of automated unit testing frameworks has become essential. This document explores Jest and other popular unit testing frameworks, comparing their features, advantages, and use cases.

**2.** Jest is a JavaScript testing framework maintained by Meta (formerly Facebook). It is widely used for testing React applications but is versatile enough to be used with other JavaScript libraries and frameworks.

**Key Features of Jest:**

* Zero configuration setup
* Snapshot testing
* Built-in code coverage
* Fast and parallel test execution
* Mocking capabilities
* Easy integration with Babel, TypeScript, and Node.js

In our project, we utilized Jest to ensure the accuracy and reliability of key components in both the frontend and backend systems. The frontend, built using React Native, relied on Jest to test user interface components, input validation functions, and integration with QR code generation modules. Meanwhile, the backend, built with Node.js, used Jest to test API endpoints, authentication logic, and utility functions.

We chose Jest because of its simple setup, snapshot testing capabilities, and built-in support for mock functions, which allowed us to isolate and verify logic effectively. This was particularly helpful when testing the QR code functionality, where we had to confirm the correct generation of data-encoded QR images based on dynamic user inputs.

Through automated testing with Jest, we increased development efficiency and ensured that updates or refactors did not break existing features. It allowed us to maintain confidence in our codebase and rapidly identify bugs during the development process.

**Link and References:**

* Jest Official Website: <https://jestjs.io>
* Using Jest with React: <https://jestjs.io/docs/tutorial-react>
* Mock Functions: <https://jestjs.io/docs/mock-functions>
* Testing Node.js with Jest: <https://jestjs.io/docs/getting-started>

**3.** One of the core features of our application was the generation and scanning of QR codes for user authentication and quick access to services. We implemented QR code functionality using libraries such as qrcode and react-native-qrcode-svg. Jest was instrumental in validating the logic that transformed user data into valid QR formats and ensuring that the generated codes were scannable and accurate.

We also used unit tests to simulate the scanning process, validating that scanned QR data matched expected values and triggered appropriate actions within the app. This testing process was essential for delivering a robust and secure QR code-based feature, especially for use cases like attendance logging, quick registration, and secure login.