

**Project X – Research**

**Prepared by: Team BAMP**

**Sophie A. Bohol**

**Cloe Den Mabanding**

**Maureen Portillo**

**Jhomarie Avenido**

In developing Project X, we encountered several technical challenges that required us to search for solutions online, consult documentation, and test different approaches. This section presents key problems we faced and the solutions we applied to overcome them.

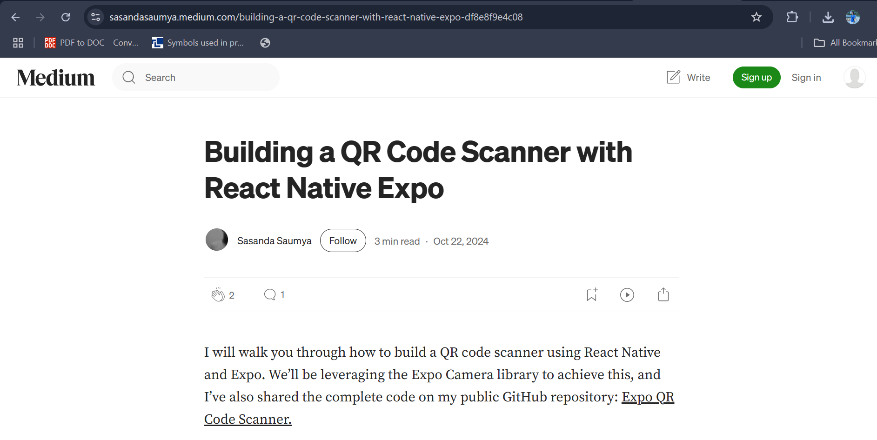
## How to generate and handle QR codes in React Native

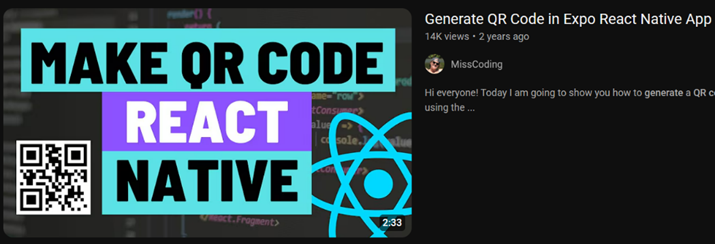
Method: Watched videos, read articles, and tested QR code libraries

**Resources:**

* https://youtu.be/iShltk7YCfM?si=8zGuFoCcd-CFrOua
* https://sasandasaumya.medium.com/building-a-qr-code-scanner-with-react-native-expo-df8e8f9e4c08

**Solution:** Used `react-native-qrcode-svg` for generating QR codes and `expo-barcode-scanner` for scanning. Implemented QR data validation and expiration handling using JSON structure.

**Outcome:** Successfully implemented dynamic QR generation and scanning with expiration validation in the attendance feature.



## 

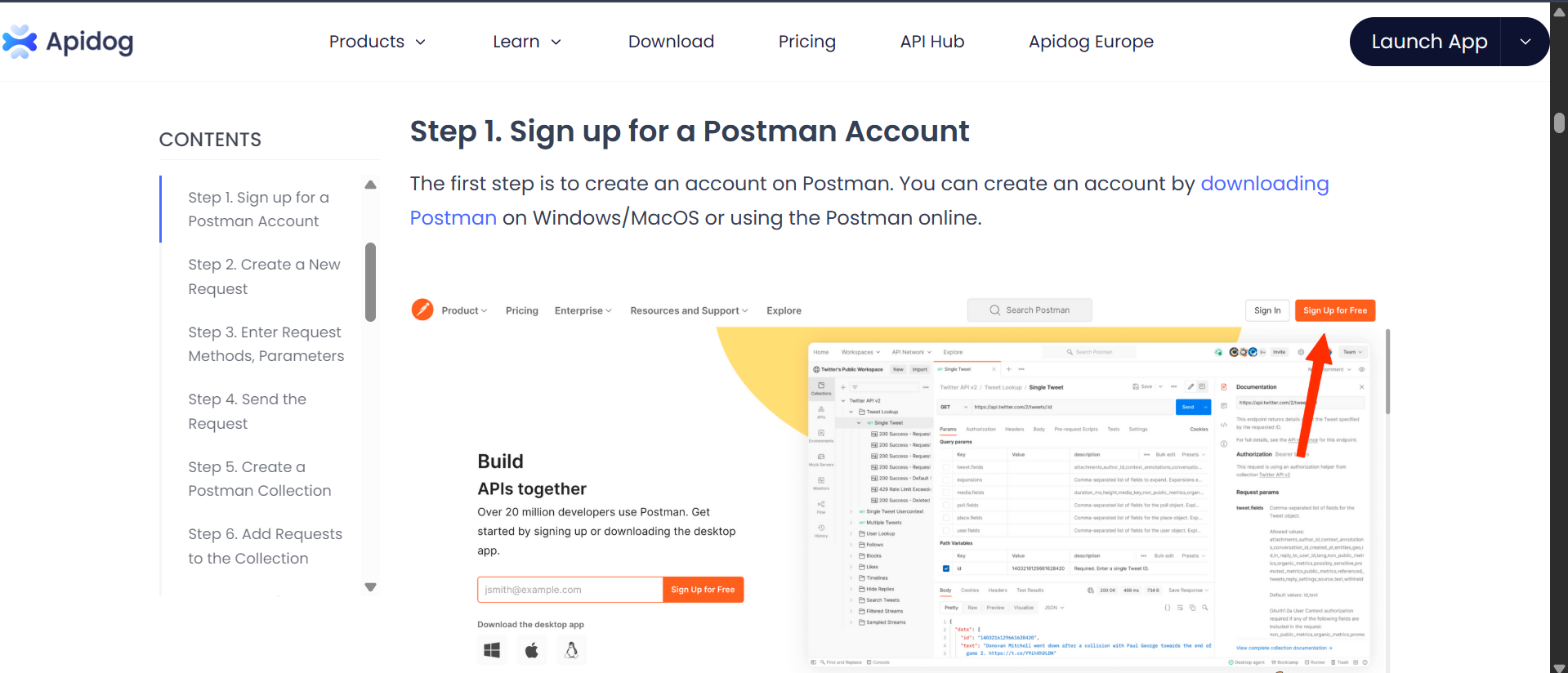
## How to test backend API endpoints using Postman

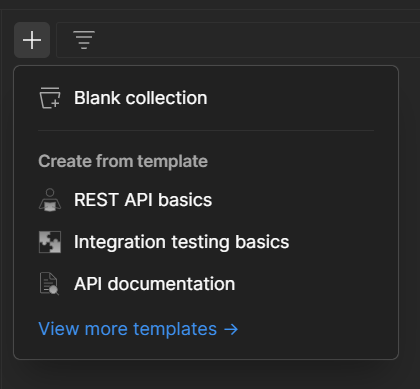
Method: Practiced using Postman with CRUD requests and researched official docs

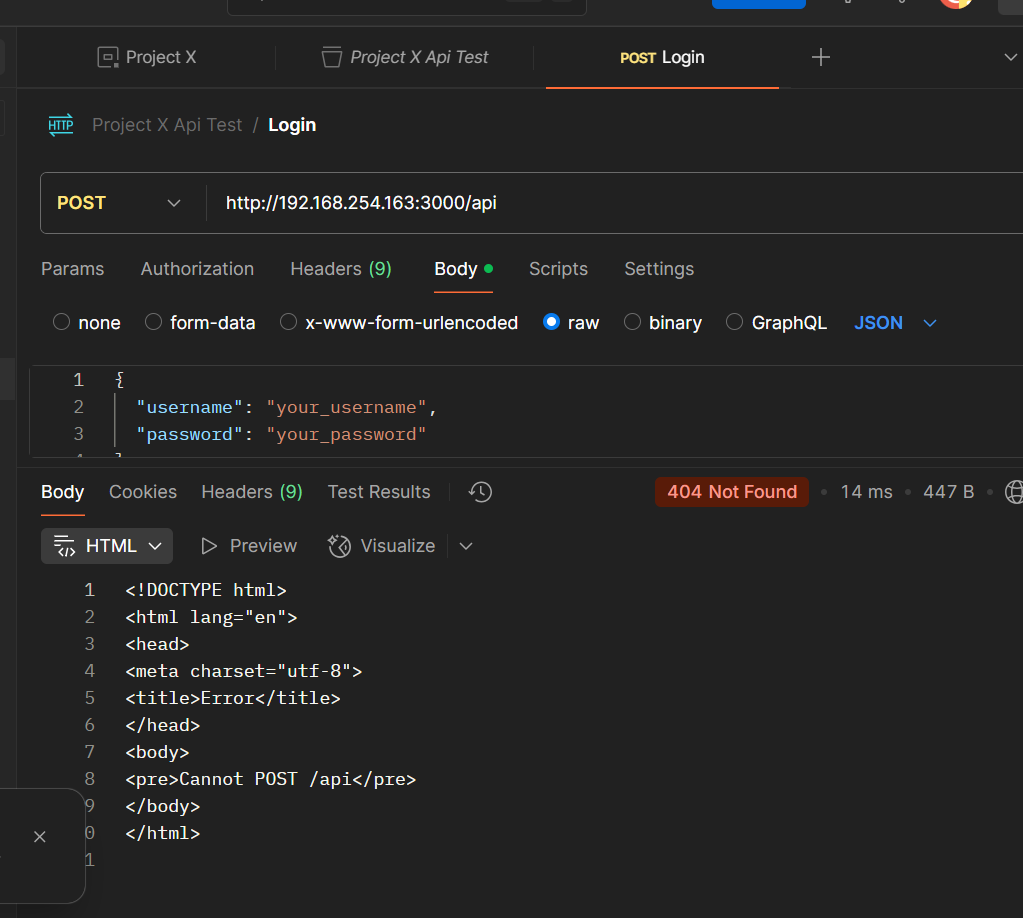
Resources:

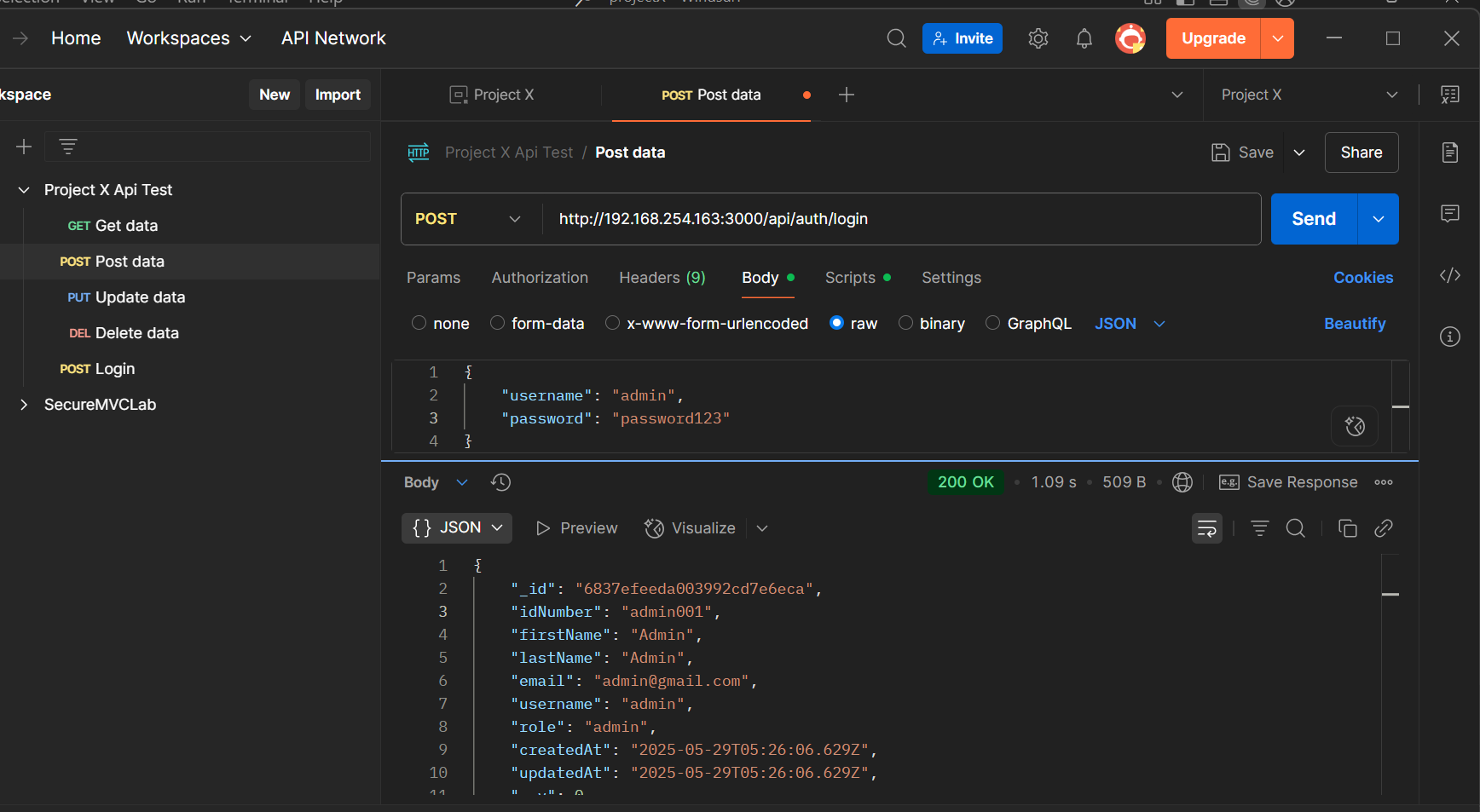
* <https://learning.postman.com/docs/getting-started/introduction/>
* https://apidog.com/blog/how-to-use-postman-for-api-testing/?utm\_source=google\_dsa&utm\_medium=g&utm\_campaign=22058259794&utm\_content=174196802804&utm\_term=&gad\_source=1&gad\_campaignid=22058259794&gbraid=0AAAAA-gKXrAmFf5JsQxuZlJS2A-SJtqEf&gclid=Cj0KCQjw0erBBhDTARIsAKO8iqTdoxksTnT6MlC4E63qeaR7wns42v5GnoZ4EV9AqxhMTS2khEhZULMaAsxIEALw\_wcB

**Solution:** Tested /api/users, /api/auth/login, and /api/courses endpoints using POST, GET, and DELETE methods to ensure correct responses and validation.

**Outcome:** Validated that all API routes return expected responses and handle invalid input correctly.







## How to perform unit testing and TDD using Jest

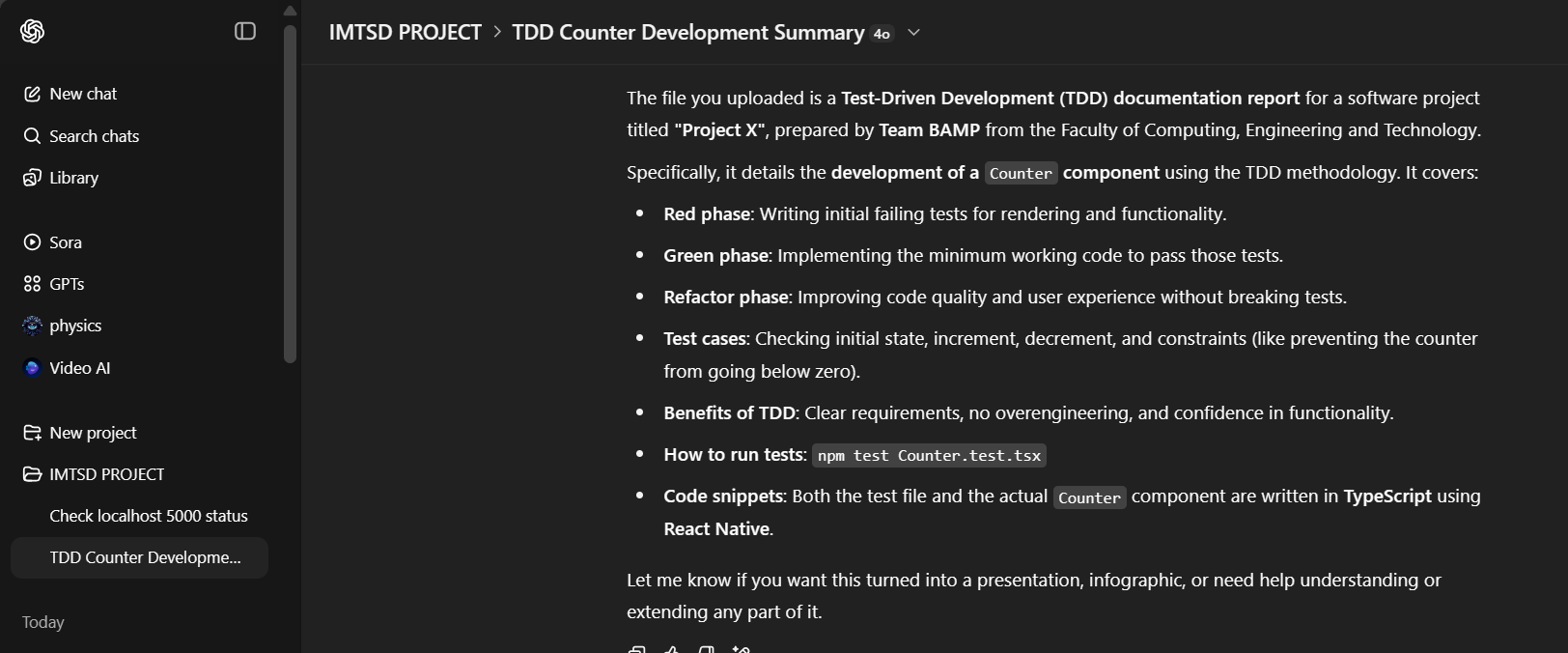
Method: Watched Jest tutorials and followed test-driven development practice

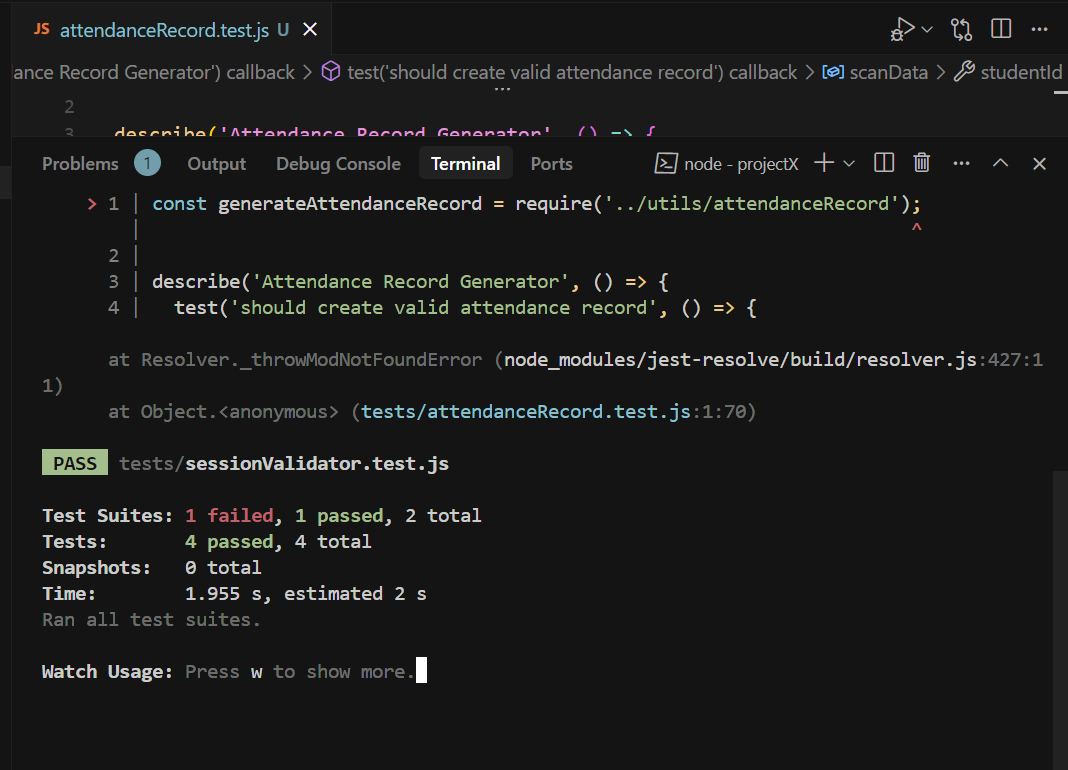
**Resources:**

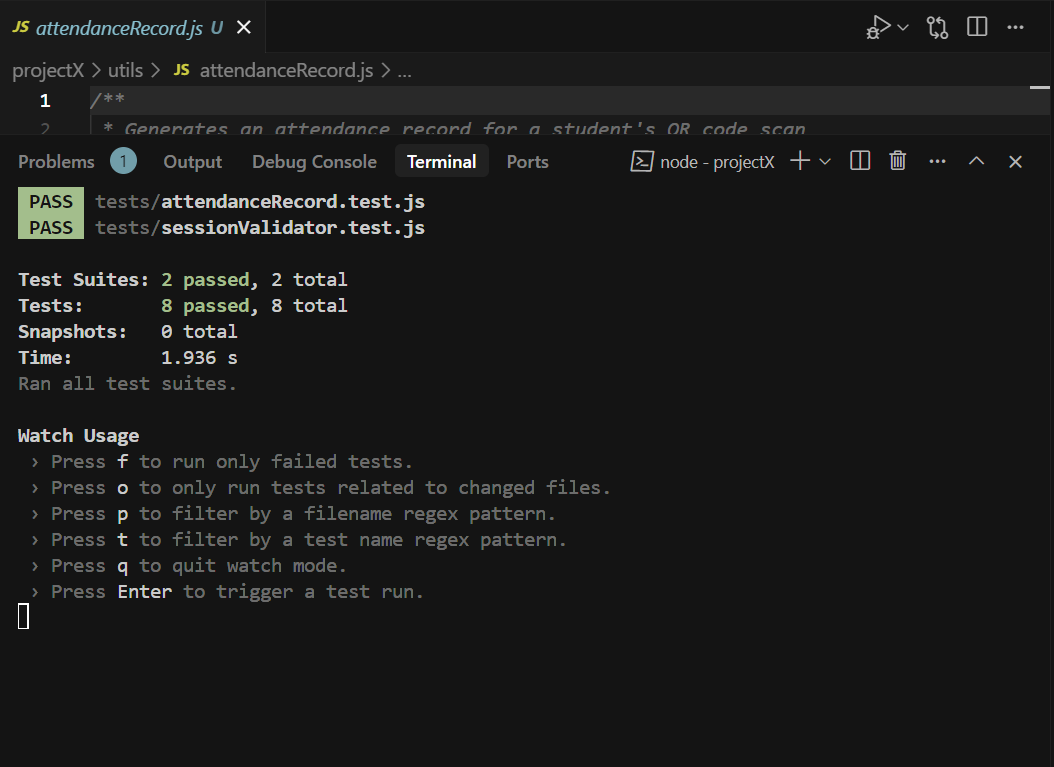
* <https://jestjs.io/docs/getting-started>
* https://chatgpt.com

**Solution:** Created tests before and after implementation using Jest and Supertest. Validated course creation, login functionality, and QR generation.

**Outcome:** Achieved test coverage for main features and improved code reliability through automated tests.







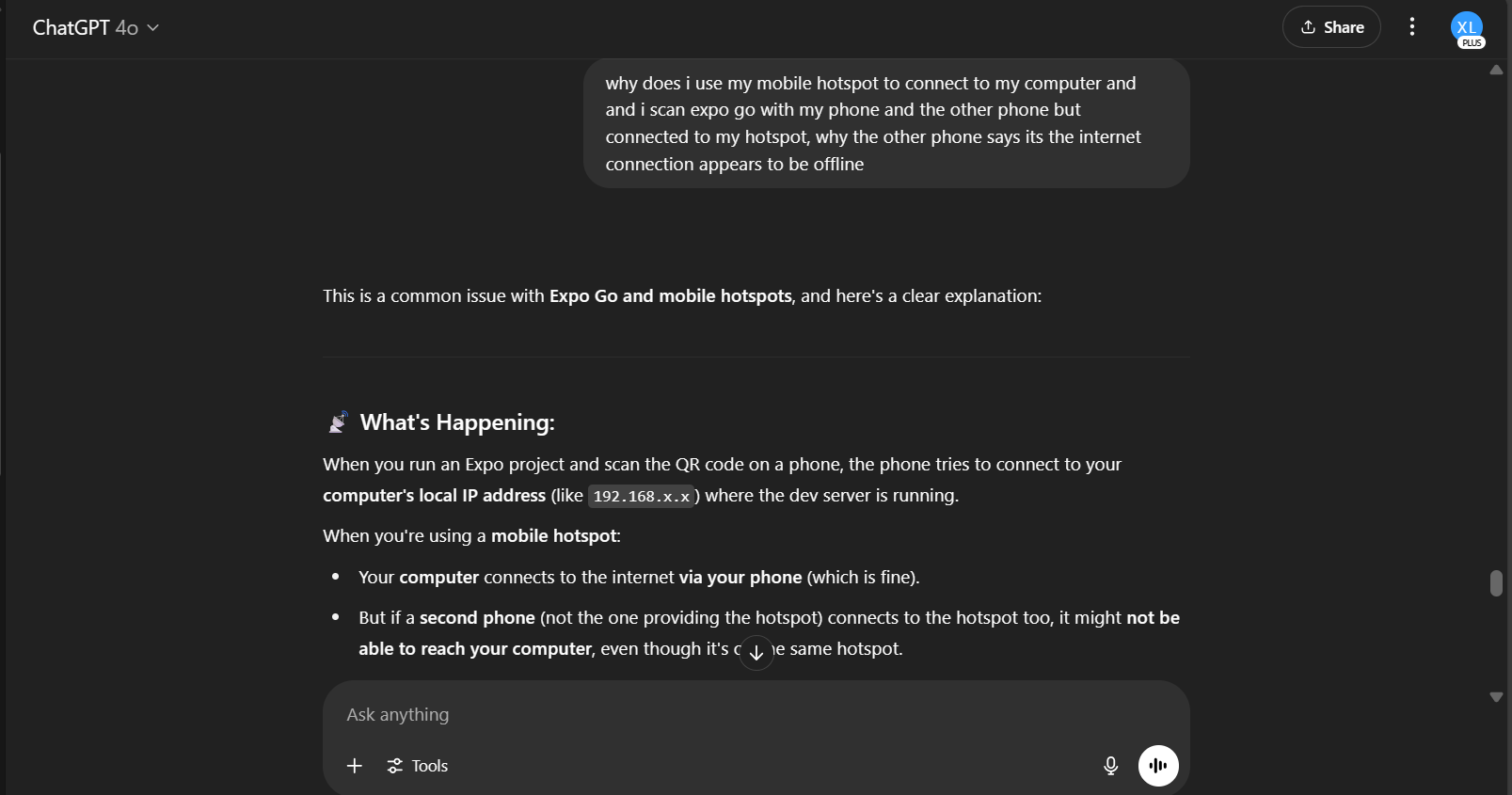
**Expo Go QR scan issue when using mobile hotspot**

Method: Asked ChatGPT, tested scenarios with multiple devices

**Resources:**

* https://chatgpt.com

**Solution:** Learned that devices on the same hotspot might not communicate due to isolation. Switched to LAN Wi-Fi or ensured same physical device for server and scanning.

**Outcome:** QR scanning worked successfully after adjusting network setup. Problem was understood and resolved.

**List of Dependencies**

**Frontend (React Native + Expo)**

npm install react react-native

npm install expo expo-font expo-router

npm install react-native-qrcode-svg

npm install expo-barcode-scanner

npm install axios

**UI libraries**

npm install react-native-svg

npm install @expo/vector-icons

npm install react-native-modal

**Backend (Node.js + Express + MongoDB)**

npm install express

npm install mongoose

npm install bcryptjs

npm install jsonwebtoken

npm install cors

**Testing (Jest + Supertest)**

npm install jest --save-dev

npm install supertest --save-dev

npm install mongodb-memory-server --save-dev

We used the following dependencies to implement key features in our system such as authentication, attendance QR, API testing, and data validation.

Throughout this project, we encountered real-world technical challenges in handling QR sessions, implementing full CRUD, and debugging network issues with Expo. Applying TDD and unit testing gave us a deeper understanding of writing clean, testable code. The hands-on experience helped us grow our skills in backend validation, API testing, and collaborative Git workflows.