

Project X - Test-Driven Development (TDD)

Prepared by: Team BAMP

Sophie A. Bohol

Cloe Den Mabanding

Maureen Portillo

Jhomarie Avenido

**Test-Driven Development (TDD)**

**QR Attendance System Features**

**1. Session Validator Implementation**

**RED Phase**

First, we wrote the tests before implementation:

```javascript

// tests/sessionValidator.test.js

const validateQRSession = require('../utils/sessionValidator');

describe('QR Session Validator', () => {

test('should throw error for invalid session format', () => {

const invalidSession = {

courseId: '123' // Missing required fields

};

expect(() => validateQRSession(invalidSession))

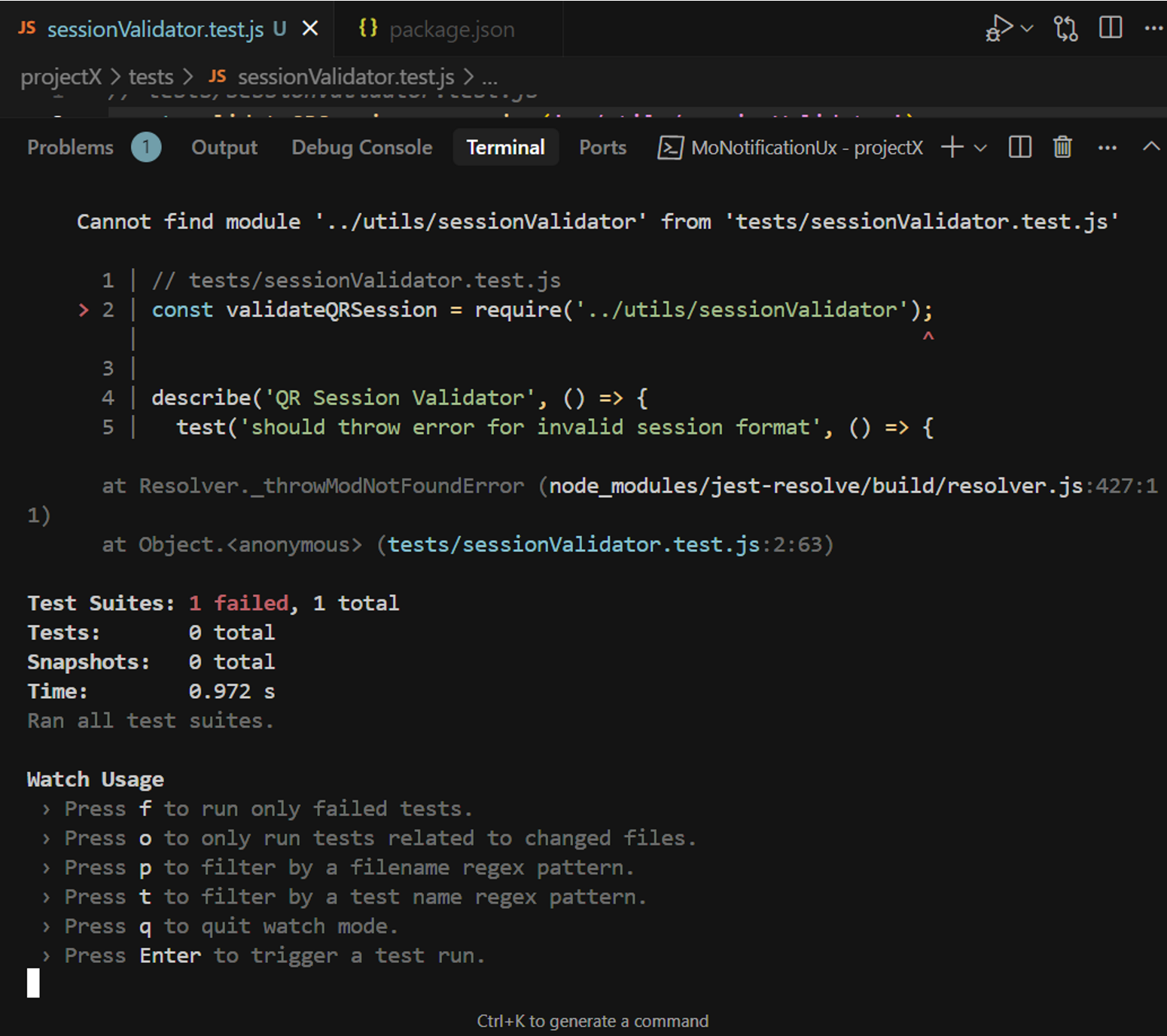
.toThrow('Invalid session format');

});

// ... other test cases

});

```



**GREEN Phase**

**We implemented the basic functionality:**

```javascript

// utils/sessionValidator.js

function validateQRSession(session) {

if (!session.generatedAt || !session.expiresAt ||

!session.courseId || !session.studentId ||

session.isEnrolled === undefined) {

throw new Error('Invalid session format');

}

if (new Date() > new Date(session.expiresAt)) {

return {

isValid: false,

message: "QR code has expired",

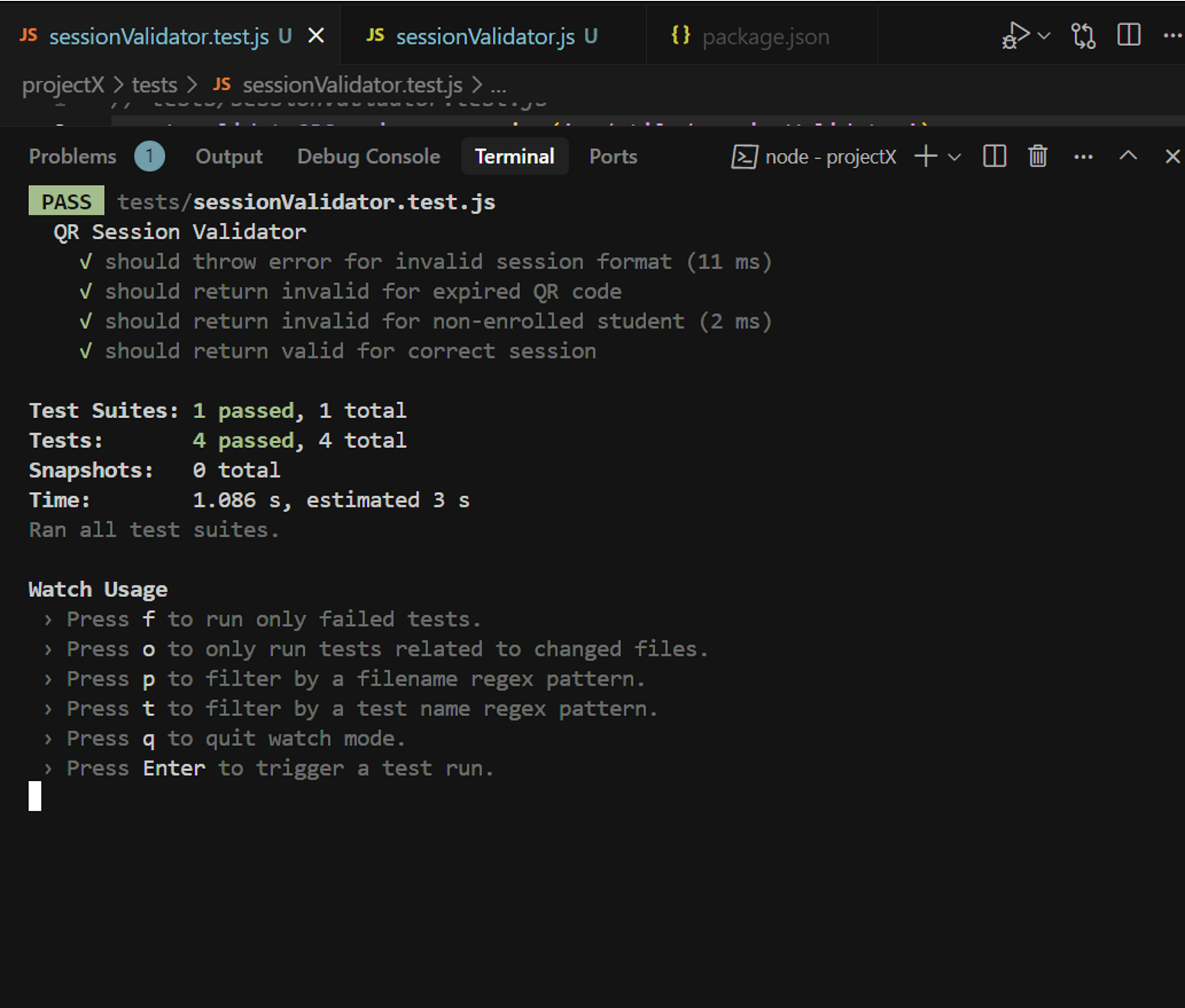
canScan: false

};

}

// ... rest of implementation

}

```

**REFACTOR Phase**

**We improved the code with better organization and documentation:**

```javascript

/\*\*

\* Validates a QR session for attendance tracking

\* @param {Object} session - The session object to validate

\* @returns {Object} Validation result

\*/

function validateQRSession(session) {

const requiredFields = ['generatedAt', 'expiresAt', 'courseId', 'studentId'];

const missingFields = requiredFields.filter(field => !session[field]);

if (missingFields.length > 0 || session.isEnrolled === undefined) {

throw new Error('Invalid session format');

}

const createResponse = (isValid, message) => ({

isValid,

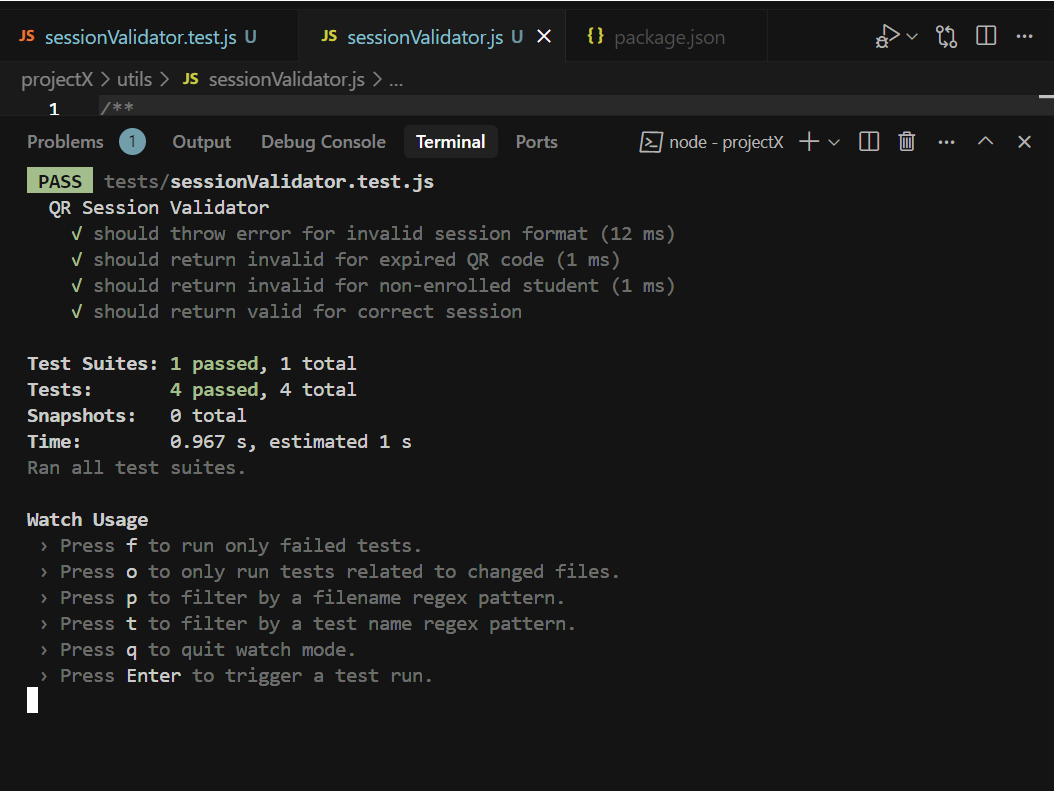
message,

canScan: isValid

});

// ... improved implementation

}

```

**2. Attendance Record Generator**

**RED Phase**

**We started with tests:**

```javascript

// tests/attendanceRecord.test.js

describe('Attendance Record Generator', () => {

test('should create valid attendance record', () => {

const scanData = {

studentId: '6836bb49c535ef45281bc334',

courseId: '6836bc3bc535ef45281bc34c',

scannedAt: new Date().toISOString()

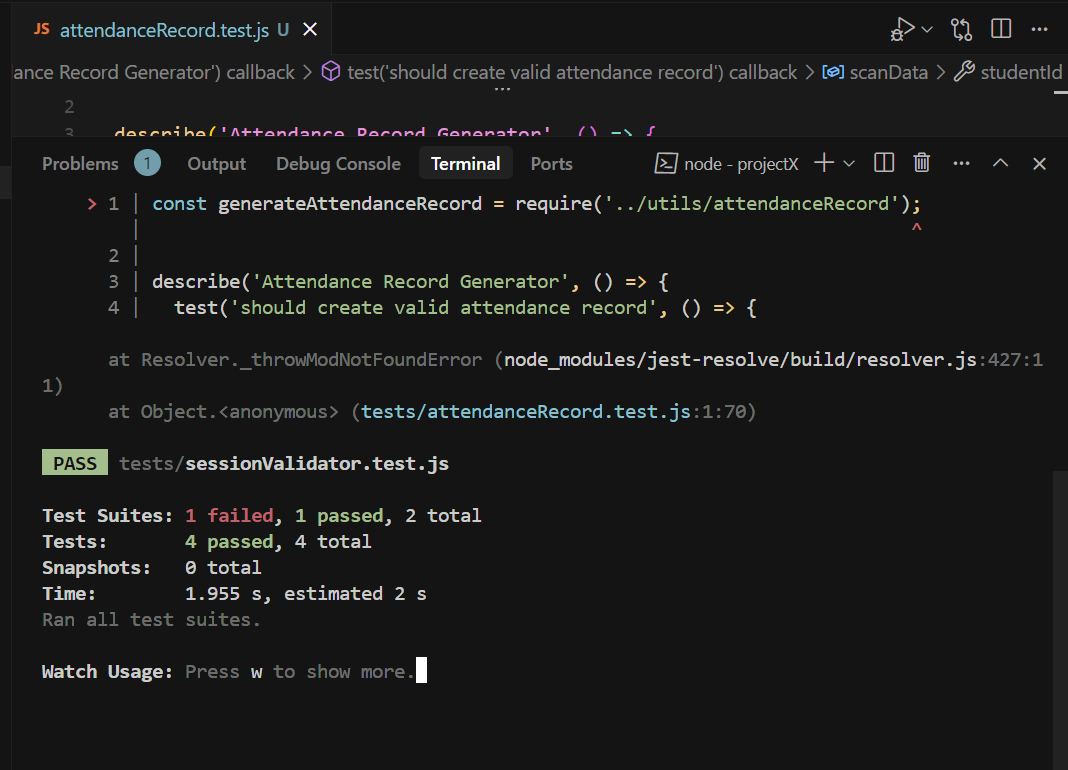
};

// ... test expectations

});

});

```



**GREEN Phase**

Basic implementation to make tests pass:

```javascript

function generateAttendanceRecord(scanData) {

if (!scanData.studentId || !scanData.courseId || !scanData.scannedAt) {

throw new Error('Missing required scan data');

}

return {

studentId: scanData.studentId,

courseId: scanData.courseId,

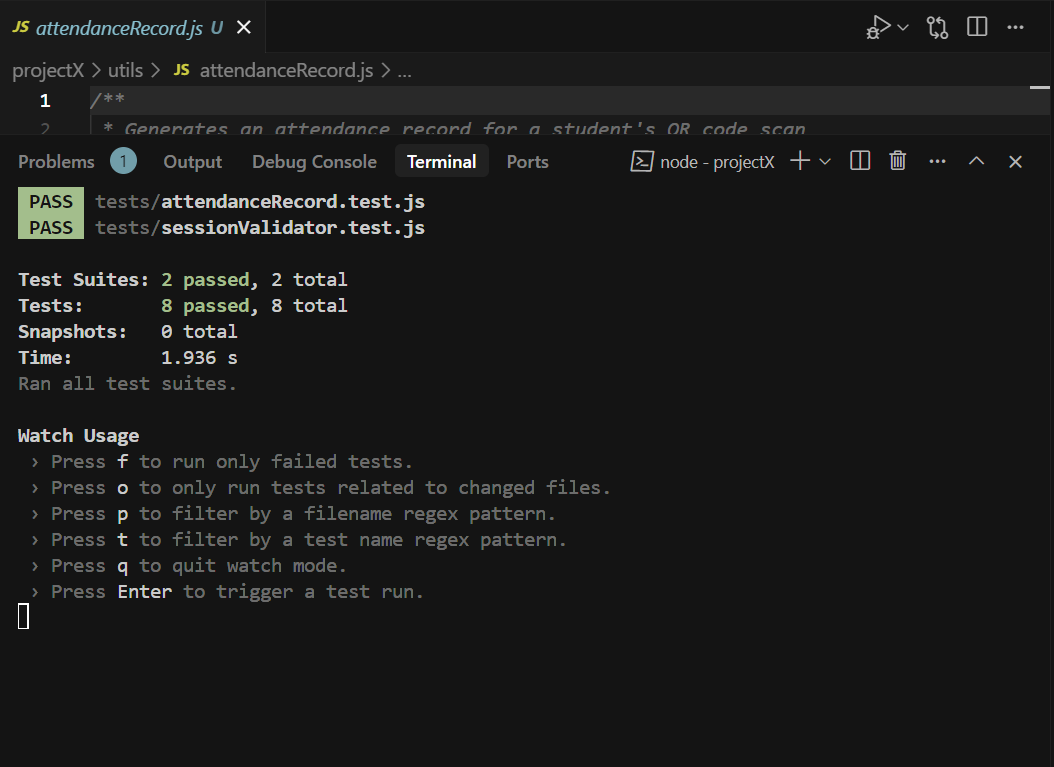
timestamp: scanData.scannedAt,

status: scanData.isLate ? 'LATE' : 'PRESENT'

};

}

```



**REFACTOR Phase**

**Enhanced with better error handling and documentation:**

```javascript

/\*\*

\* Generates an attendance record for a student's QR code scan

\* @param {Object} scanData - The data from QR code scan

\* @throws {Error} If required scan data is missing

\*/

function generateAttendanceRecord(scanData) {

const requiredFields = ['studentId', 'courseId', 'scannedAt'];

const missingFields = requiredFields.filter(field => !scanData[field]);

if (missingFields.length > 0) {

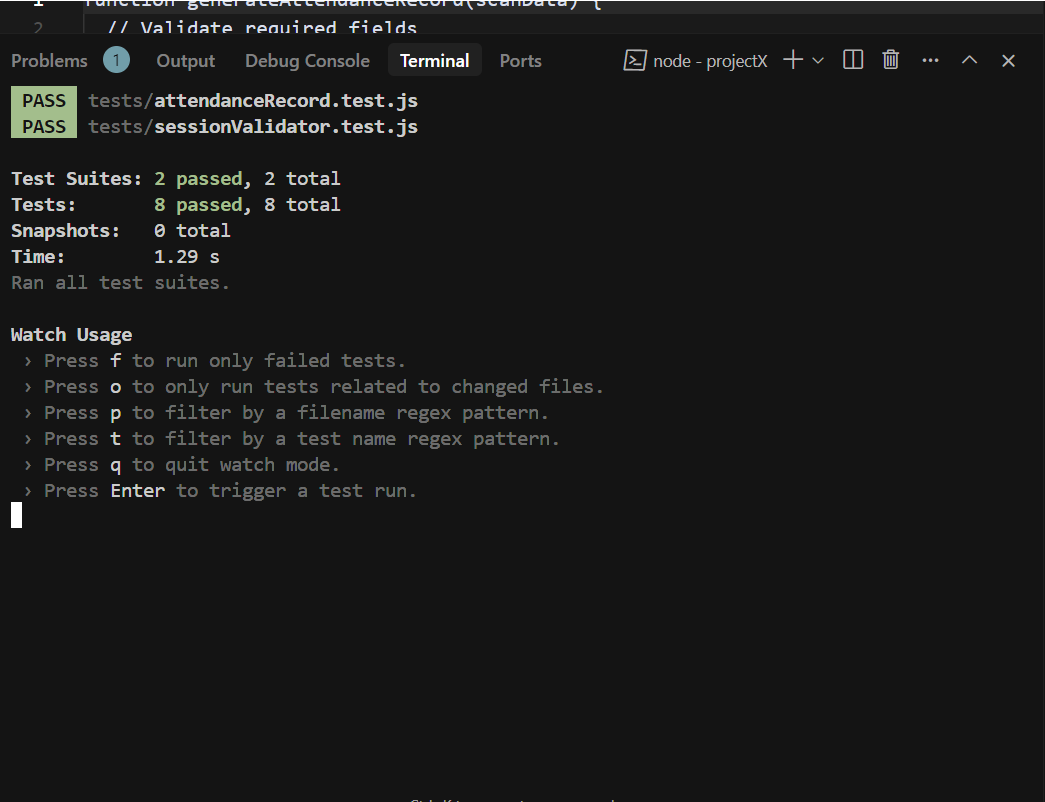
throw new Error(`Missing required scan data: ${missingFields.join(', ')}`);

}

// ... improved implementation

}

```



**Test Coverage:**

- Session validation: 4 test cases

- Attendance records: 4 test cases

- Total: 8 passing tests

**Running the Tests:**

```bash

npm test tests/sessionValidator.test.js

npm test tests/attendanceRecord.test.js

```

```