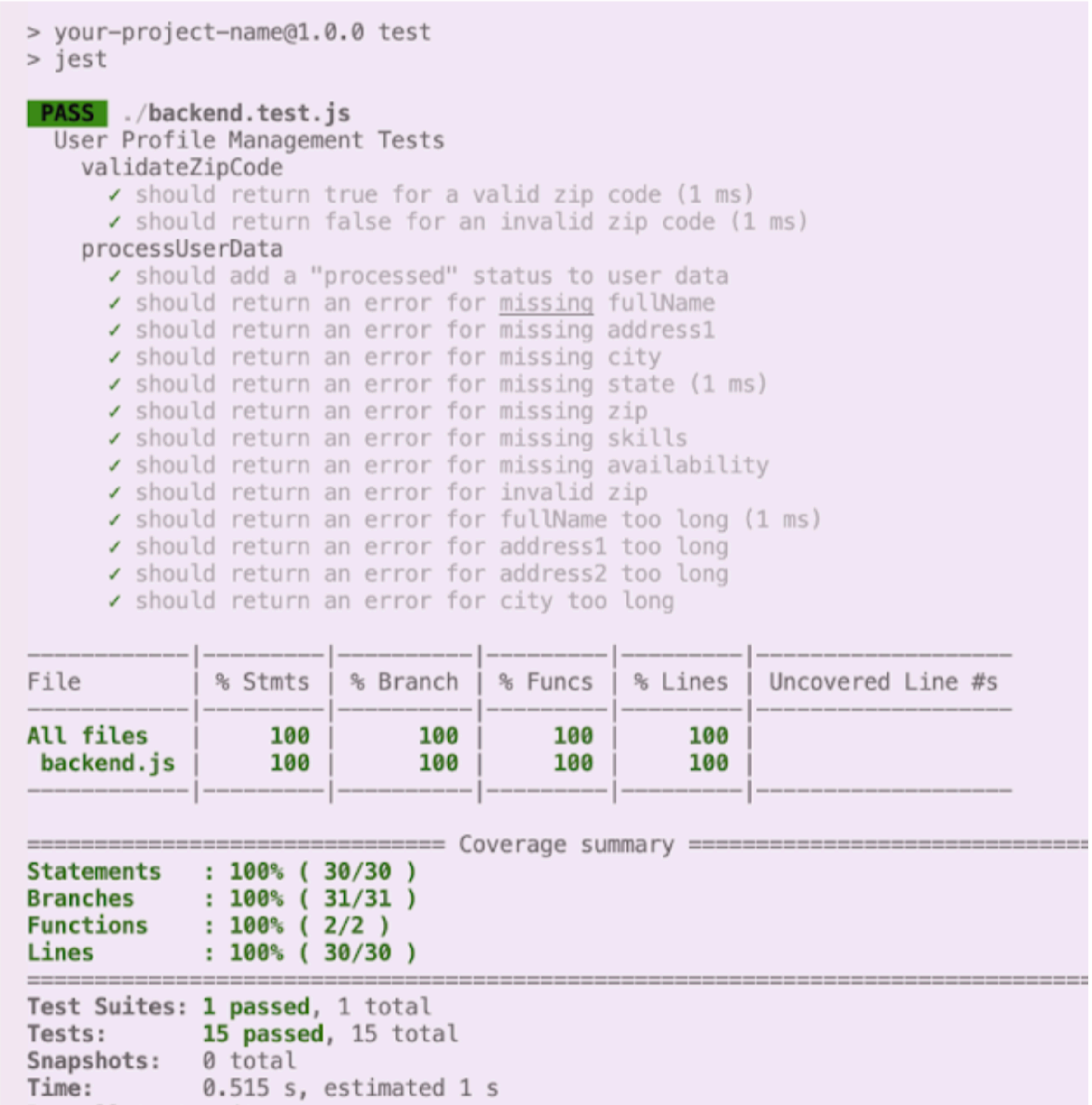
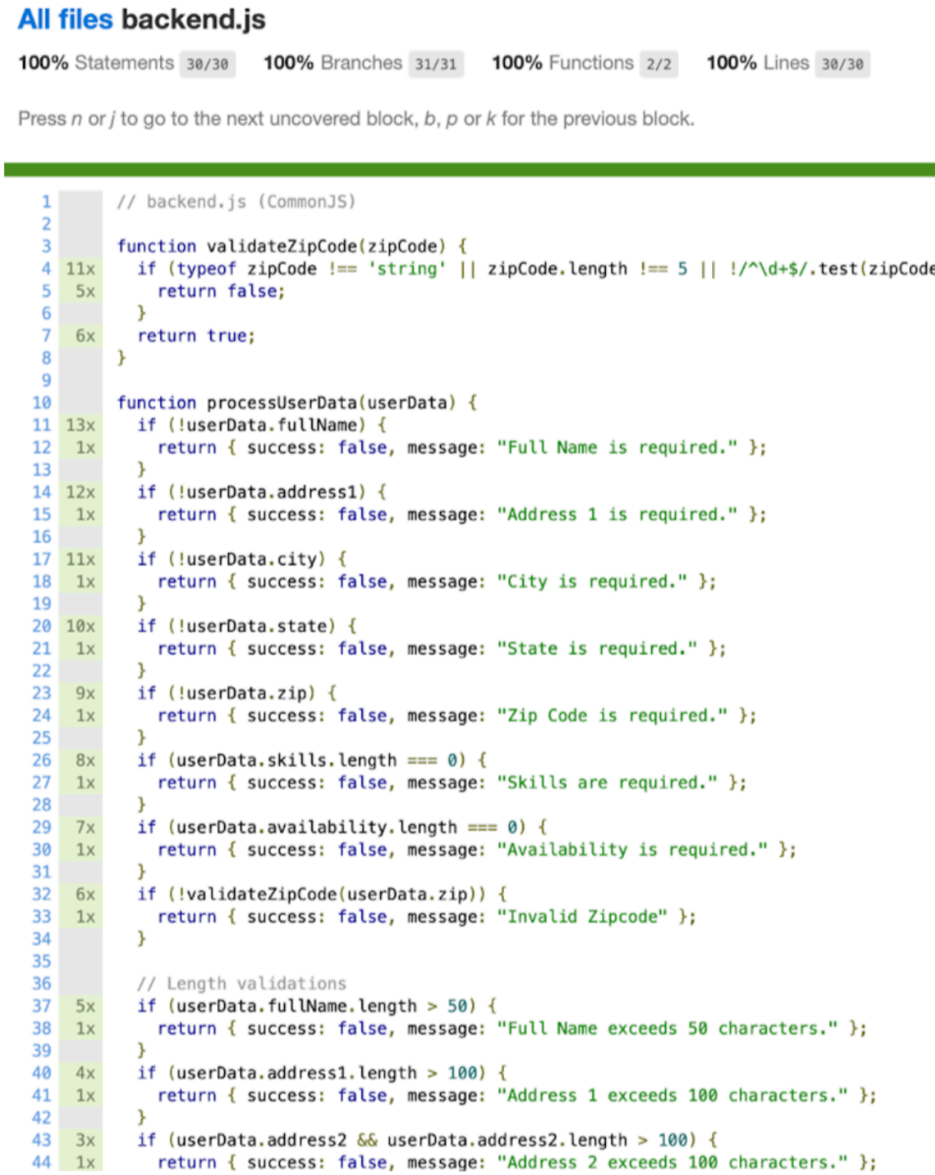


```
> your-project-name@1.0.0 test
> jest

 PASS  ./backend.test.js
  User Profile Management Tests
    validateZipCode
      ✓ should return true for a valid zip code (1 ms)
      ✓ should return false for an invalid zip code (1 ms)
    processUserData
      ✓ should add a "processed" status to user data
      ✓ should return an error for missing fullName
      ✓ should return an error for missing address1
      ✓ should return an error for missing city
      ✓ should return an error for missing state (1 ms)
      ✓ should return an error for missing zip
      ✓ should return an error for missing skills
      ✓ should return an error for missing availability
      ✓ should return an error for invalid zip
      ✓ should return an error for fullName too long (1 ms)
      ✓ should return an error for address1 too long
      ✓ should return an error for address2 too long
      ✓ should return an error for city too long

------------|---------|----------|---------|---------|-------------------
File        | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------|---------|----------|---------|---------|-------------------
All files   |     100 |      100 |     100 |     100 |
 backend.js |     100 |      100 |     100 |     100 |
------------|---------|----------|---------|---------|-------------------

=============================== Coverage summary ===============================
Statements   : 100% ( 30/30 )
Branches     : 100% ( 31/31 )
Functions    : 100% ( 2/2 )
Lines        : 100% ( 30/30 )
================================================================================
Test Suites: 1 passed, 1 total
Tests:       15 passed, 15 total
Snapshots:   0 total
Time:        0.515 s, estimated 1 s
```



All files backend.js

100% Statements 30/30 **100%** Branches 31/31 **100%** Functions 2/2 **100%** Lines 30/30

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

```
     // backend.js (CommonJS)

     function validateZipCode(zipCode) {
11x    if (typeof zipCode !== 'string' || zipCode.length !== 5 || !/^\d+$/.test(zipCode
 5x      return false;
       }
 6x    return true;
     }

     function processUserData(userData) {
13x    if (!userData.fullName) {
 1x      return { success: false, message: "Full Name is required." };
       }
12x    if (!userData.address1) {
 1x      return { success: false, message: "Address 1 is required." };
       }
11x    if (!userData.city) {
 1x      return { success: false, message: "City is required." };
       }
10x    if (!userData.state) {
 1x      return { success: false, message: "State is required." };
       }
 9x    if (!userData.zip) {
 1x      return { success: false, message: "Zip Code is required." };
       }
 8x    if (userData.skills.length === 0) {
 1x      return { success: false, message: "Skills are required." };
       }
 7x    if (userData.availability.length === 0) {
 1x      return { success: false, message: "Availability is required." };
       }
 6x    if (!validateZipCode(userData.zip)) {
 1x      return { success: false, message: "Invalid Zipcode" };
       }

       // Length validations
 5x    if (userData.fullName.length > 50) {
 1x      return { success: false, message: "Full Name exceeds 50 characters." };
       }
 4x    if (userData.address1.length > 100) {
 1x      return { success: false, message: "Address 1 exceeds 100 characters." };
       }
 3x    if (userData.address2 && userData.address2.length > 100) {
 1x      return { success: false, message: "Address 2 exceeds 100 characters." };
```