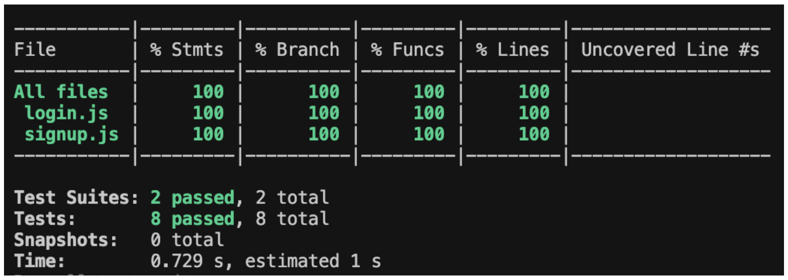
COSC 4353 Assignment 3 -- Group19

1. **GitHub Repository Link** (5 points)
   - https://github.com/ElliotMadsen/COSC4353---Software-Design
   - Note: Assignment 3 is split up into branches for each section of the web app and denoted with **-Assignment3** at the end
2. **Backend Technologies** (2 points)
   - JavaScript: We are most familiar with JavaScript and found it had a variety of accessible resources to aid us as we worked.
   - Jest: We used Jest for its code coverage features, as this was the most straightforward method in combination with JS and VS Code.
3. **Team Contributions** (3 points)

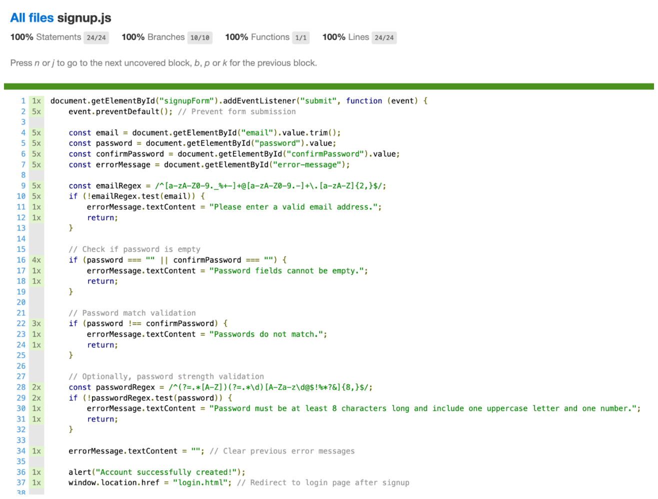
Group Member Name	What is your contribution?	Discussion Notes
1. Elliot	Login/Signup Module	Includes frontend, backend, and code coverage tests for this module.
2. Elena	User Profile Management Module	Includes frontend, backend, and code coverage tests for this module.
3. Prakash	Notification System/Volunteer History Modules	Includes frontend, backend, and code coverage tests for this module.
4. Wendy	Event Management/Volunteer Matching Modules	Includes frontend, backend, and code coverage tests for this module.

Code Coverage Test Results (also included in Github Repository)

- Login/Signup



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

Test Suites: 2 passed, 2 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        0.729 s, estimated 1 s
```

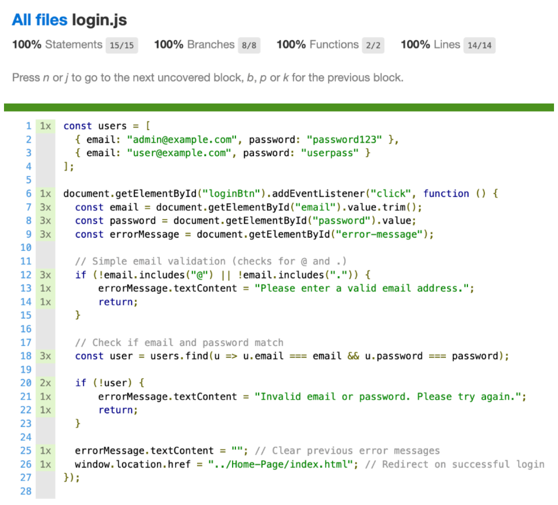


All files signup.js

100% Statements 24/24 100% Branches 10/10 100% Functions 1/1 100% Lines 24/24

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

```
1x  document.getElementById("signupForm").addEventListener("submit", function (event) {
5x      event.preventDefault(); // Prevent form submission

5x      const email = document.getElementById("email").value.trim();
5x      const password = document.getElementById("password").value;
5x      const confirmPassword = document.getElementById("confirmPassword").value;
5x      const errorMessage = document.getElementById("error-message");

5x      const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
5x      if (!emailRegex.test(email)) {
1x          errorMessage.textContent = "Please enter a valid email address.";
1x          return;
        }

        // Check if password is empty
4x      if (password === "" || confirmPassword === "") {
1x          errorMessage.textContent = "Password fields cannot be empty.";
1x          return;
        }

        // Password match validation
3x      if (password !== confirmPassword) {
1x          errorMessage.textContent = "Passwords do not match.";
1x          return;
        }

        // Optionally, password strength validation
2x      const passwordRegex = /^(?=.*[A-Z])(?=.*\d)[A-Za-z\d@$!%*?&]{8,}$/;
2x      if (!passwordRegex.test(password)) {
1x          errorMessage.textContent = "Password must be at least 8 characters long and include one uppercase letter and one number.";
1x          return;
        }

1x      errorMessage.textContent = ""; // Clear previous error messages

1x      alert("Account successfully created!");
1x      window.location.href = "login.html"; // Redirect to login page after signup
```

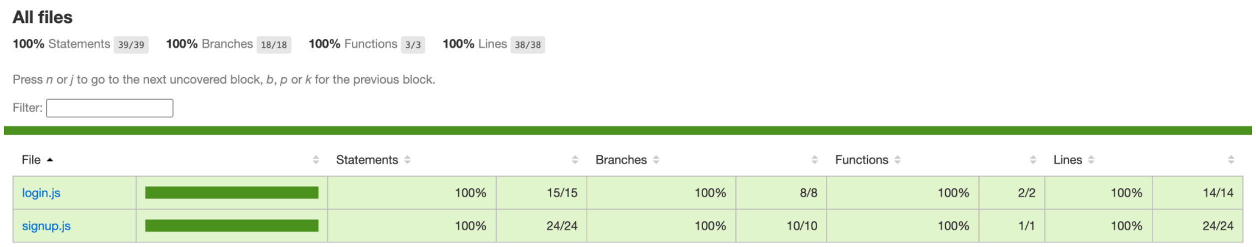


All files login.js

100% Statements 15/15 100% Branches 8/8 100% Functions 2/2 100% Lines 14/14

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

```
1x  const users = [
        { email: "admin@example.com", password: "password123" },
        { email: "user@example.com", password: "userpass" }
    ];

1x  document.getElementById("loginBtn").addEventListener("click", function () {
3x      const email = document.getElementById("email").value.trim();
3x      const password = document.getElementById("password").value;
3x      const errorMessage = document.getElementById("error-message");

        // Simple email validation (checks for @ and .)
3x      if (!email.includes("@") || !email.includes(".")) {
1x          errorMessage.textContent = "Please enter a valid email address.";
1x          return;
        }

        // Check if email and password match
3x      const user = users.find(u => u.email === email && u.password === password);

2x      if (!user) {
1x          errorMessage.textContent = "Invalid email or password. Please try again.";
1x          return;
        }

1x      errorMessage.textContent = ""; // Clear previous error messages
1x      window.location.href = "../Home-Page/index.html"; // Redirect on successful login
    });
```



All files

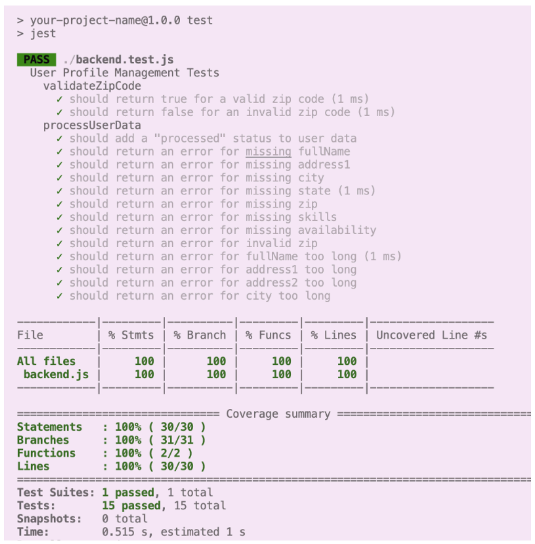
100% Statements 39/39 100% Branches 18/18 100% Functions 3/3 100% Lines 38/38

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

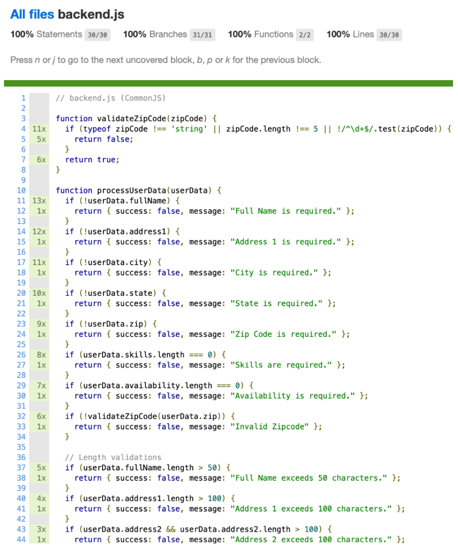
Filter:

File ▲		Statements		Branches		Functions		Lines	
login.js	🟩	100%	15/15	100%	8/8	100%	2/2	100%	14/14
signup.js	🟩	100%	24/24	100%	10/10	100%	1/1	100%	24/24

- User Profile Management



```
> 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
```

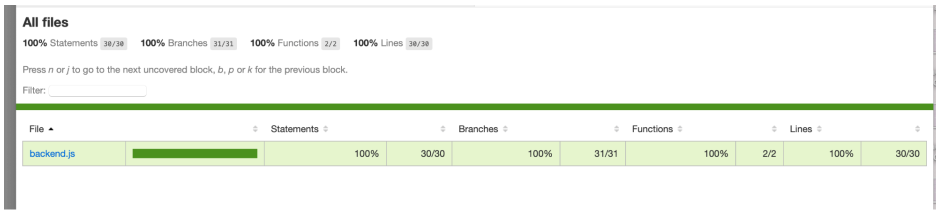


All files backend.js

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." };
```



All files

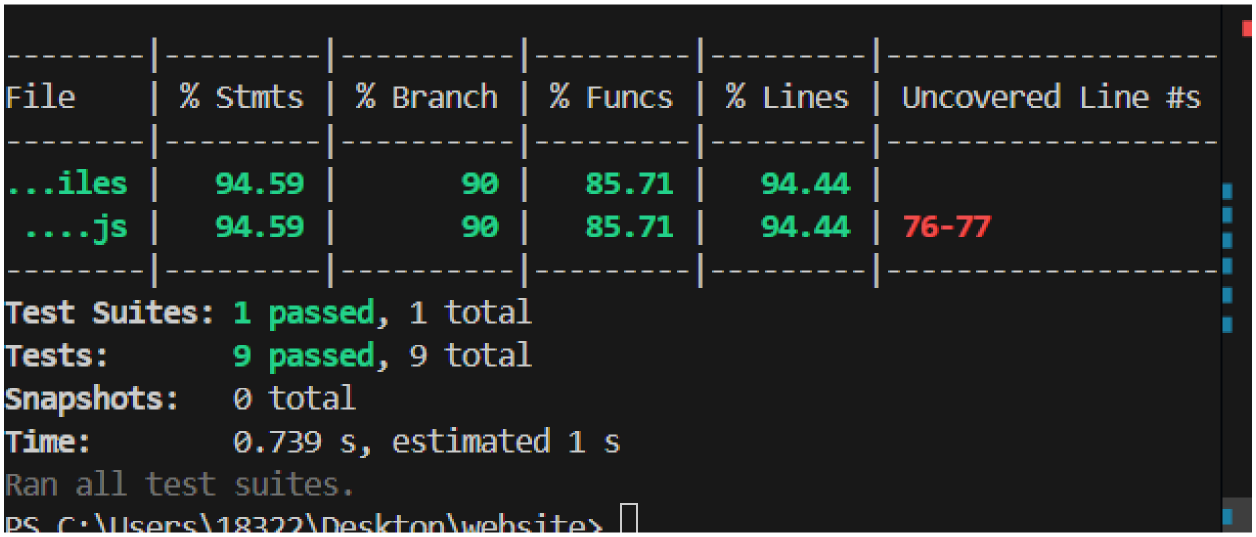
100% Statements 30/30 100% Branches 31/31 100% Functions 2/2 100% Lines 30/30

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

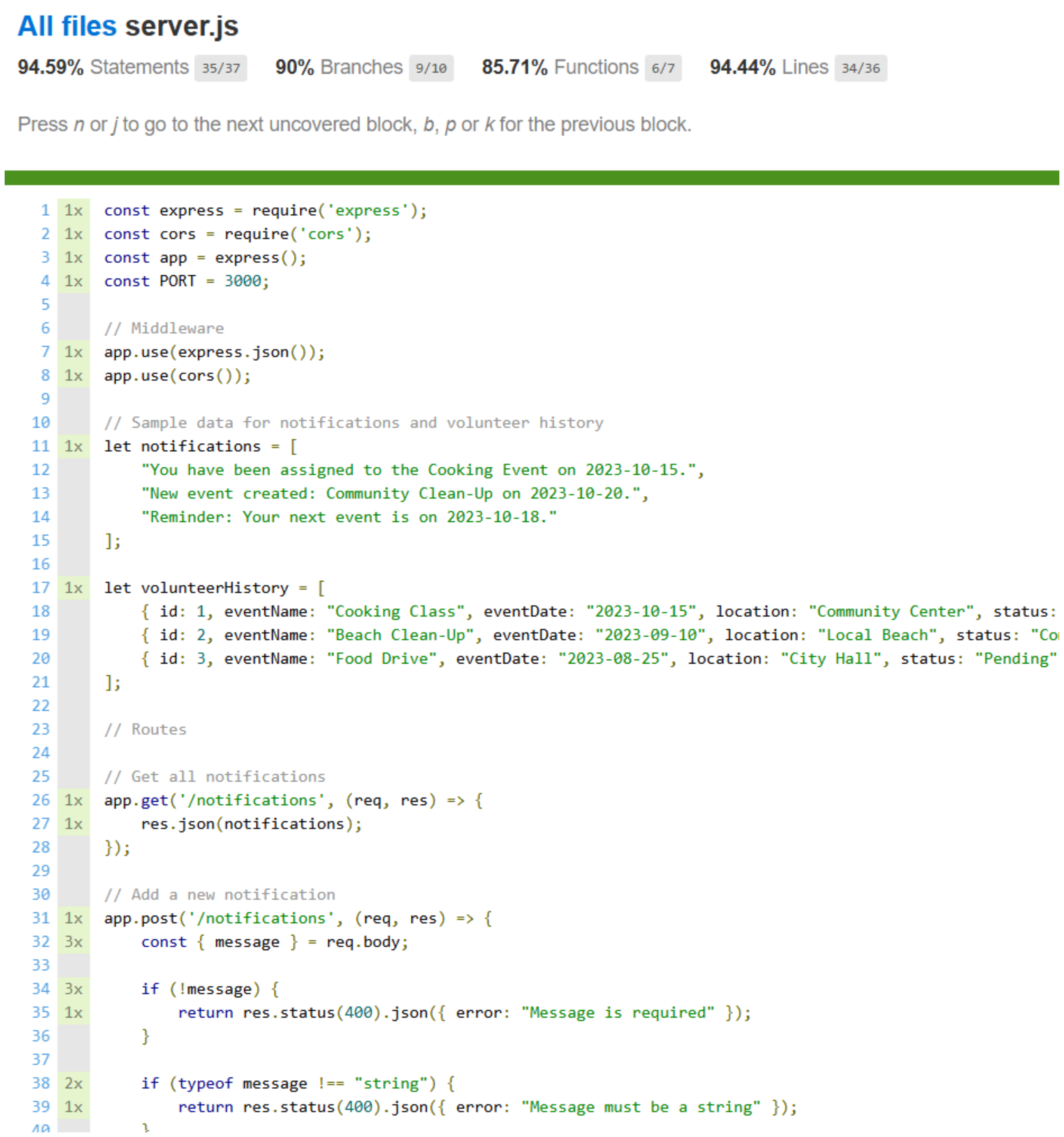
Filter:

File ▲		Statements		Branches		Functions		Lines	
backend.js	🟩	100%	30/30	100%	31/31	100%	2/2	100%	30/30

- Notification System/Volunteer History



```
--------|---------|----------|---------|---------|-------------------
File    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--------|---------|----------|---------|---------|-------------------
...iles |   94.59 |       90 |   85.71 |   94.44 |
 ....js |   94.59 |       90 |   85.71 |   94.44 | 76-77
--------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        0.739 s, estimated 1 s
Ran all test suites.
PS C:\Users\18322\Desktop\website>
```

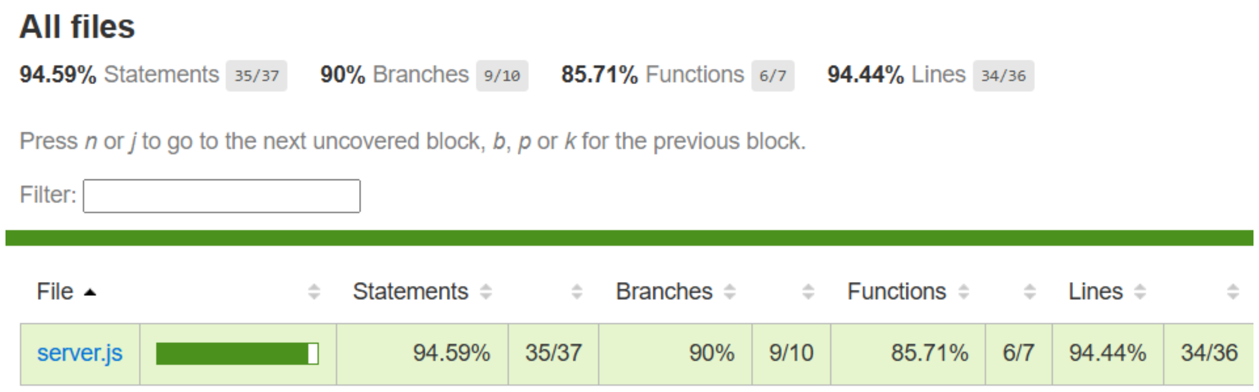


All files server.js

94.59% Statements 35/37   **90%** Branches 9/10   **85.71%** Functions 6/7   **94.44%** Lines 34/36

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

```
1x  const express = require('express');
1x  const cors = require('cors');
1x  const app = express();
1x  const PORT = 3000;

    // Middleware
1x  app.use(express.json());
1x  app.use(cors());

    // Sample data for notifications and volunteer history
1x  let notifications = [
        "You have been assigned to the Cooking Event on 2023-10-15.",
        "New event created: Community Clean-Up on 2023-10-20.",
        "Reminder: Your next event is on 2023-10-18."
    ];

1x  let volunteerHistory = [
        { id: 1, eventName: "Cooking Class", eventDate: "2023-10-15", location: "Community Center", status:
        { id: 2, eventName: "Beach Clean-Up", eventDate: "2023-09-10", location: "Local Beach", status: "Co
        { id: 3, eventName: "Food Drive", eventDate: "2023-08-25", location: "City Hall", status: "Pending"
    ];

    // Routes

    // Get all notifications
1x  app.get('/notifications', (req, res) => {
1x      res.json(notifications);
    });

    // Add a new notification
1x  app.post('/notifications', (req, res) => {
3x      const { message } = req.body;

3x      if (!message) {
1x          return res.status(400).json({ error: "Message is required" });
        }

2x      if (typeof message !== "string") {
1x          return res.status(400).json({ error: "Message must be a string" });
        }
```



All files

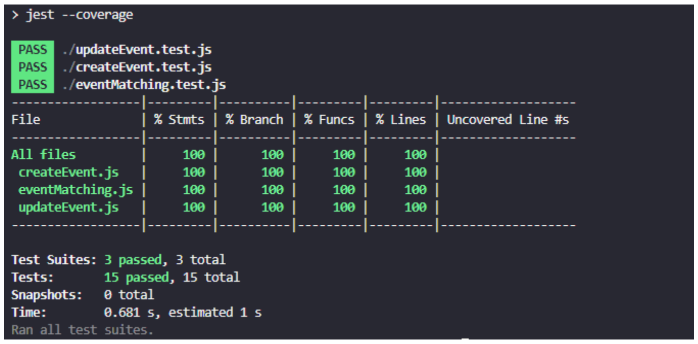
94.59% Statements 35/37 **90%** Branches 9/10 **85.71%** Functions 6/7 **94.44%** Lines 34/36

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

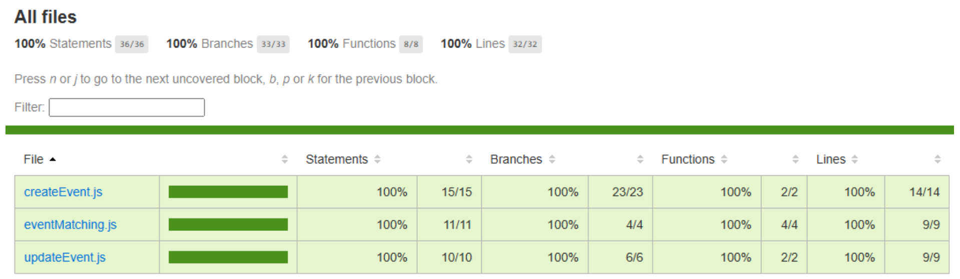
Filter:

File ▲		Statements		Branches		Functions		Lines	
server.js		94.59%	35/37	90%	9/10	85.71%	6/7	94.44%	34/36

- Event Management/Volunteer Matching



```
> jest --coverage

 PASS  ./updateEvent.test.js
 PASS  ./createEvent.test.js
 PASS  ./eventMatching.test.js
------------------|---------|----------|---------|---------|-------------------
File              | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------------|---------|----------|---------|---------|-------------------
All files         |     100 |      100 |     100 |     100 |
 createEvent.js   |     100 |      100 |     100 |     100 |
 eventMatching.js |     100 |      100 |     100 |     100 |
 updateEvent.js   |     100 |      100 |     100 |     100 |
------------------|---------|----------|---------|---------|-------------------

Test Suites: 3 passed, 3 total
Tests:       15 passed, 15 total
Snapshots:   0 total
Time:        0.681 s, estimated 1 s
Ran all test suites.
```



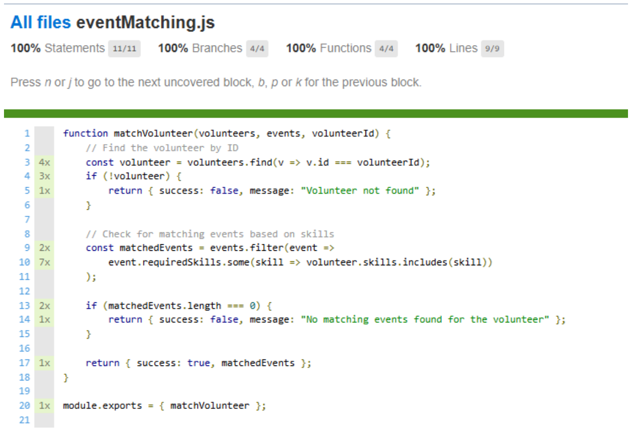
All files

100% Statements 36/36 **100%** Branches 33/33 **100%** Functions 8/8 **100%** Lines 32/32

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

Filter:

File ▲		Statements		Branches		Functions		Lines	
createEvent.js		100%	15/15	100%	23/23	100%	2/2	100%	14/14
eventMatching.js		100%	11/11	100%	4/4	100%	4/4	100%	9/9
updateEvent.js		100%	10/10	100%	6/6	100%	2/2	100%	9/9

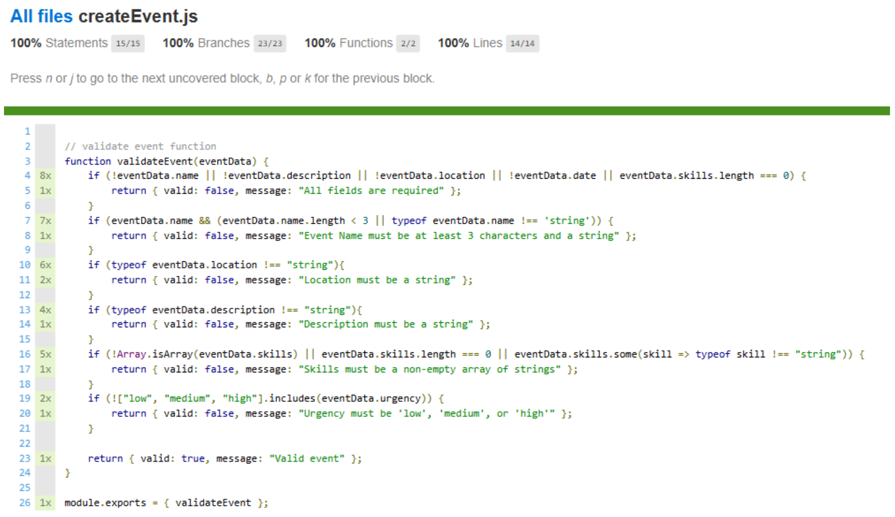


All files eventMatching.js

100% Statements 11/11 **100%** Branches 4/4 **100%** Functions 4/4 **100%** Lines 9/9

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

```
     function matchVolunteer(volunteers, events, volunteerId) {
         // Find the volunteer by ID
4x       const volunteer = volunteers.find(v => v.id === volunteerId);
3x       if (!volunteer) {
1x           return { success: false, message: "Volunteer not found" };
         }

         // Check for matching events based on skills
2x       const matchedEvents = events.filter(event =>
7x           event.requiredSkills.some(skill => volunteer.skills.includes(skill))
         );

2x       if (matchedEvents.length === 0) {
1x           return { success: false, message: "No matching events found for the volunteer" };
         }

1x       return { success: true, matchedEvents };
     }

1x   module.exports = { matchVolunteer };
```

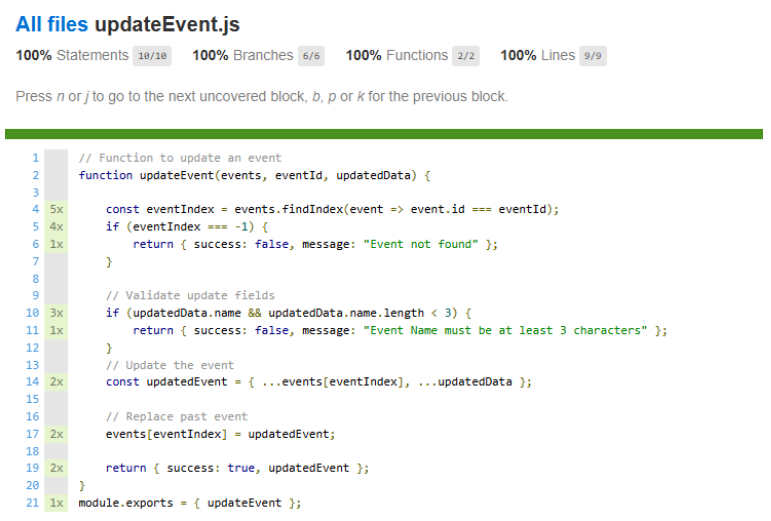


All files createEvent.js

100% Statements 15/15 **100%** Branches 23/23 **100%** Functions 2/2 **100%** Lines 14/14

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

```

     // validate event function
     function validateEvent(eventData) {
8x       if (!eventData.name || !eventData.description || !eventData.location || !eventData.date || eventData.skills.length === 0) {
1x           return { valid: false, message: "All fields are required" };
         }
7x       if (eventData.name && (eventData.name.length < 3 || typeof eventData.name !== 'string')) {
1x           return { valid: false, message: "Event Name must be at least 3 characters and a string" };
         }
6x       if (typeof eventData.location !== "string"){
2x           return { valid: false, message: "Location must be a string" };
         }
4x       if (typeof eventData.description !== "string"){
1x           return { valid: false, message: "Description must be a string" };
         }
5x       if (!Array.isArray(eventData.skills) || eventData.skills.length === 0 || eventData.skills.some(skill => typeof skill !== "string")) {
1x           return { valid: false, message: "Skills must be a non-empty array of strings" };
         }
2x       if (!["low", "medium", "high"].includes(eventData.urgency)) {
1x           return { valid: false, message: "Urgency must be 'low', 'medium', or 'high'" };
         }

1x       return { valid: true, message: "Valid event" };
     }

1x   module.exports = { validateEvent };
```



All files updateEvent.js

100% Statements 10/10 **100%** Branches 6/6 **100%** Functions 2/2 **100%** Lines 9/9

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

```
    // Function to update an event
    function updateEvent(events, eventId, updatedData) {

5x      const eventIndex = events.findIndex(event => event.id === eventId);
4x      if (eventIndex === -1) {
1x          return { success: false, message: "Event not found" };
        }

        // Validate update fields
3x      if (updatedData.name && updatedData.name.length < 3) {
1x          return { success: false, message: "Event Name must be at least 3 characters" };
        }
        // Update the event
2x      const updatedEvent = { ...events[eventIndex], ...updatedData };

        // Replace past event
2x      events[eventIndex] = updatedEvent;

2x      return { success: true, updatedEvent };
    }
1x  module.exports = { updateEvent };
```