

## **Jest**

### **Coding:**

#### **View Hospital Component Test**

```
import React from 'react';
```

```
import { render, screen, waitFor } from '@testing-library/react';
```

```
import axios from 'axios';
```

```
import ViewHospitalComponent from '../Component/ViewHospitalComponent';
```

```
jest.mock('axios');
```

```
describe('ViewHospitalComponent', () => {
```

```
  it('renders hospital data fetched from API', async () => {
```

```
    const mockData = {
```

```
      data: [
```

```
        {
```

```
          "id": "26e36c6a-0c9a-48e2-8eac-0296706b2604",
```

```
          "name": "ABC Hospital",
```

```
          "email": "smano4570@gmail.com",
```

```
          "phoneNumber": 6382202586,
```

```
          "location": "Chennai",
```

```
"governmentId": "1245433",
```

```
"document": "document",
```

```
"doorNo": "21/2",
```

```
"street": "Salai Street,",
```

```
"area": "Ktc Nagar",
```

```
"city": "Tirunelveli",
```

```
"state": "Tamil Nadu",
```

```
"postalCode": "627011",
```

```
"status": 1
```

```
}
```

```
]
```

```
}
```

```
axios.get.mockResolvedValue(mockData);
```

```
const { getByTestId, getAllByTestId } = render(<ViewHospitalComponent />);
```

```
await waitFor(() => {  
  
  expect(getByTestId('row-26e36c6a-0c9a-48e2-8eac-  
0296706b2604')).toBeInTheDocument();  
  
});  
  
});  
  
});
```

### **View Donor Component Test**

```
import React from 'react';  
  
import { render, waitFor } from '@testing-library/react';  
  
import axios from 'axios';  
  
import ViewDonorComponent from '../Component/ViewDonorComponent';
```

```
jest.mock('axios');
```

```
describe('ViewDonorComponent', () => {  
  
  it('renders donor information', async () => {  
  
    const data = {
```

```
data: [
{
  "accountUserDetailsAddressId": "50fa9fae-bfb2-4e5c-a219-2ebdc1fb4f47",
  "account": {
    "accountId": "477b7d70-299c-4428-a130-7b5bf5176586",
    "name": "Keerthi Vasan",
    "email": "keerthivasan280901@gmail.com",
    "password":
      "ceb200a6cea4095b86c8bc4d1e801aff9812eaa106fd940e348882d0268d59f5",
    "phoneNumber": 9089678776,
    "status": 1,
    "bloodTransaction": null,
    "bloodCampBloodBank": null,
    "bloodBankBloodStock": null
  },
  "userDetails": {
    "userDetailsId": "1b60eba4-66f9-496f-a764-7afb10d7ee87",
    "age": 21,
    "bloodType": "B+ve",
    "location": "Chennai",
    "governmentId": "",
    "document": "",
    "aadhaarNumber": 345623458900,
```

```
"rolestatus": 3

},

"address": {

"addressId": "b6872d86-1eb1-4fca-b2ad-3f2c798440b7",

"doorNo": "21",

"street": "Vk Street",

"area": "Narayana nagar",

"city": "Perambalur",

"state": "Tamil Nadu",

"postalCode": "627001"

}

}

]

};
```

```
axios.get.mockResolvedValue(data);
```

```
const { getByTestId } = render(<ViewDonorComponent />);

await waitFor(() => {

expect(getByTestId('row-477b7d70-299c-4428-a130-
7b5bf5176586')).toBeInTheDocument();
```

```
});
```

```
});
```

```
});
```

### **View Blood Transaction Component Test**

```
import React from 'react';
```

```
import { render, waitFor } from '@testing-library/react';
```

```
import axios from 'axios';
```

```
import ViewBloodTransaction from '../Component/ViewBloodTransaction';
```

```
import Cookies from 'js-cookie';
```

```
jest.mock('axios');
```

```
jest.mock('js-cookie');
```

```
describe('ViewBloodTransaction', () => {
```

```
  it('renders blood transaction information', async () => {
```

```
    const data = {
```

```
    data: [
```

```
    {
```

```
      "bloodTransactionId": "cbb7f7ec-2ebf-45b3-af88-12729cb4fa41",
```

```
      "accountId": "10bab100-d6b3-41d0-9d3b-46491cda7ca0",
```

```
"account": {  
  
  "accountId": "10bab100-d6b3-41d0-9d3b-46491cda7ca0",  
  
  "name": "ABC Blood Bank",  
  
  "email": "sanjaihari2002@gmail.com",  
  
  "password":  
    "ceb200a6cea4095b86c8bc4d1e801aff9812eaa106fd940e348882d0268d59f5",  
  
  "phoneNumber": 6382202487,  
  
  "status": 1,  
  
  "bloodTransaction": null,  
  
  "bloodCampBloodBank": null,  
  
  "bloodBankBloodStock": null  
},  
  
  "bloodRequestId": "T2937kCU",  
  
  "bloodRequest": {  
  
    "bloodRequestId": "T2937kCU",  
  
    "name": "Sanjai",  
  
    "email": "sanjairock85@gmail.com",  
  
    "units": 2,  
  
    "phoneNumber": 9867541243,  
  
    "bloodType": "B+ve",  
  
    "age": 20,  
  
    "location": "Chennai",  
  
    "aadhaarNumber": 235489098762,
```

```
"validTime": "03/29/2024 17:11:20",  
  
"status": 4,  
  
"acceptStatus": 1,  
  
"bloodTransaction": null  
  
},  
  
"bloodType": "B+ve",  
  
"units": 0,  
  
"date": "string",  
  
"time": "string"  
  
}  
  
]  
  
};
```

```
axios.post.mockResolvedValue(data);  
  
Cookies.get.mockReturnValue('sample-id');
```

```
const { getByTestId } = render(<ViewBloodTransaction />);  
  
await waitFor(() => {  
  
  expect(getByTestId('row-cbb7f7ec-2ebf-45b3-af88-  
12729cb4fa41')).toBeInTheDocument();  
  
});
```



```
});
```

```
});
```

### **Unauthorized Component Test**

```
import React from 'react';
```

```
import { render } from '@testing-library/react';
```

```
import UnauthorizedComponent from '../Component/UnauthorizedComponent';
```

```
describe('UnauthorizedComponent', () => {
```

```
  it('renders without crashing', () => {
```

```
    const { getByTestId } = render(<UnauthorizedComponent />);
```

```
    expect(getByTestId('unauthorized-section')).toBeInTheDocument();
```

```
  });
```

```
  it('displays the unauthorized image', () => {
```

```
    const { getByTestId } = render(<UnauthorizedComponent />);
```

```
    const img = getByTestId('unauthorized-image');
```

```
    expect(img).toBeInTheDocument();
```

```
    expect(img).toHaveAttribute('src', expect.stringContaining('accessdenied.png'));
```

```
});
```

```
});
```

### **Reject Component Test**

```
import React from 'react';
```

```
import { render } from '@testing-library/react';
```

```
import RejectComponent from '../Component/RejectComponent';
```

```
describe('RejectComponent', () => {
```

```
  it('renders the reject image', () => {
```

```
    const { getByTestId } = render(<RejectComponent />);
```

```
    const imgElement = getByTestId('reject-image');
```

```
    expect(imgElement).toBeInTheDocument();
```

```
    expect(imgElement).toHaveAttribute('src', expect.stringContaining('reject.png'));
```

```
  });
```

```
});
```

### **Pending Component Test**

```
import React from 'react';
```

```
import { render, screen } from '@testing-library/react';
```

```
import { BrowserRouter } from 'react-router-dom';
```

```
import PendingComponent from '../Component/PendingComponent';
```

```
describe('PendingComponent', () => {  
  it('renders the pending message', () => {  
    render(  
      <BrowserRouter>  
        <PendingComponent />  
      </BrowserRouter>  
    );
```

```
    const pendingMessage = screen.getByText(/Your Request has been waiting for  
    management approval/i);
```

```
    expect(pendingMessage).toBeInTheDocument();  
  });
```

```
});
```

### **Not Found Component Test**

```
import React from 'react';
```

```
import { render } from '@testing-library/react';

import NotFoundComponent from '../Component/NotFoundComponent';
```

```
describe('NotFoundComponent', () => {

  it('renders without crashing', () => {

    const { getByTestId } = render(<NotFoundComponent />);

    expect(getByTestId('notfound-section')).toBeInTheDocument();

  });
```

```
    it('displays the not found image with correct dimensions', () => {

      const { getByTestId } = render(<NotFoundComponent />);

      const img = getByTestId('notfound-image');

      expect(img).toBeInTheDocument();

      expect(img).toHaveAttribute('src', expect.stringContaining('notfound.png'));

      expect(img).toHaveAttribute('width', '350px');

      expect(img).toHaveAttribute('height', '150px');

    });

  });
```

## Login Component Test

```
import React from 'react';

import { render, fireEvent, waitFor } from '@testing-library/react';

import LoginComponent from '../Component/LoginComponent';

import axios from 'axios'

import { BrowserRouter } from 'react-router-dom';
```

```
const mockedNavigate = jest.fn();

jest.mock('react-router-dom', () => ({
  ...jest.requireActual('react-router-dom'),
  useNavigate: () => mockedNavigate,
}));
```

```
describe('LoginComponent', () => {

  it('renders email and password inputs', () => {

    const { getByTestId } = render(

      <BrowserRouter><LoginComponent /></BrowserRouter>;

    const emailInput = getByTestId('email-input');

    const passwordInput = getByTestId('password-input');
```

```
expect(emailInput).toBeInTheDocument();  
expect(passwordInput).toBeInTheDocument();  
});
```

```
it('submits form with valid email and password', async () => {  
  const mockAxios = jest.spyOn(axios, 'post').mockResolvedValueOnce({  
    data: {  
      accountExists: true,  
      passwordStatus: true,  
      accountApproval: 'approve',  
      token: 'mockToken',  
    },  
  });
```

```
const { getByTestId } = render(<BrowserRouter><LoginComponent /></BrowserRouter>);  
const emailInput = getByTestId('email-input');  
const passwordInput = getByTestId('password-input');
```

```
const loginButton = getByTestId('login-button');

fireEvent.change(emailInput, { target: { value: 'validemail@example.com' } });

fireEvent.change(passwordInput, { target: { value: 'validpassword' } });

fireEvent.click(loginButton);

await waitFor(() => {

  expect(mockAxios).toHaveBeenCalledWith('https://localhost:7089/api/Login', {

    Email: 'validemail@example.com',

    Password: 'validpassword',

  });

});

});

});

});
```

### **Forget Password Test**

```
import React from 'react';

import { render, fireEvent, waitFor } from '@testing-library/react';

import ForgetPasswordComponent from '../Component/ForgetPasswordComponent';

import axios from 'axios';

import { BrowserRouter } from 'react-router-dom';
```

```
jest.mock('axios');

describe('ForgetPasswordComponent', () => {

  it('renders correctly', () => {

    const { getByTestId } = render(

      <BrowserRouter>

      <ForgetPasswordComponent />

    </BrowserRouter>

  );

  expect(getByTestId('email-input')).toBeInTheDocument();

  expect(getByTestId('submit-button')).toBeInTheDocument();

});

it('submits the form with valid email', async () => {

  axios.post.mockResolvedValue({ data: { emailExists: true, sendMail: true } });

  const { getByTestId } = render(

    <BrowserRouter>

    <ForgetPasswordComponent />

  </BrowserRouter>

);

  fireEvent.change(getByTestId('email-input'), { target: { value: 'test@example.com' } });

  fireEvent.click(getByTestId('submit-button'));

  await waitFor(() => {

    expect(axios.post).toHaveBeenCalledWith('https://localhost:7089/api/ForgetPassword', {

      email: 'test@example.com',
```



```
});
```

```
});
```

```
});
```

```
});
```

### **Change Password Test**

```
import React from 'react';
```

```
import { render, fireEvent, waitFor } from '@testing-library/react';
```

```
import ChangePasswordComponent from '../Component/ChangePasswordComponent';
```

```
import axios from 'axios';
```

```
import { BrowserRouter as Router } from 'react-router-dom';
```

```
jest.mock('axios');
```

```
const mockedNavigate = jest.fn();
```

```
jest.mock('react-router-dom', () => ({
```

```
...jest.requireActual('react-router-dom'),
```

```
useNavigate: () => mockedNavigate,
```

```
}));
```

```
describe('ChangePasswordComponent', () => {
```

```
it('renders correctly', () => {
```

```
const { getByTestId } = render(<Router>
```

```
<ChangePasswordComponent />
```

```
</Router>);
```

```
expect(getByTestId('email-input')).toBeInTheDocument();
```

```
expect(getByTestId('old-password-input')).toBeInTheDocument();
```

```

expect(getByTestId('new-password-input')).toBeInTheDocument();

expect(getByTestId('submit-button')).toBeInTheDocument();

});

it('submits the form with valid data', async () => {

  axios.post.mockResolvedValue({ data: { emailExists: true, passcheck: true } });

  const { getByTestId } = render(<ChangePasswordComponent />);

  fireEvent.change(getByTestId('email-input'), { target: { value: 'test@example.com' } });

  fireEvent.change(getByTestId('old-password-input'), { target: { value: 'oldPassword123' } });

  fireEvent.change(getByTestId('new-password-input'), { target: { value:
  'newPassword123' } });

  fireEvent.click(getByTestId('submit-button'));

  await waitFor(() => {

    expect(axios.post).toHaveBeenCalledWith('https://localhost:7089/api/ChangePassword',
    {

      email: 'test@example.com',

      oldPassword: 'oldPassword123',

      newPassword: 'newPassword123',

    });

  });

});

});

});

});

```

### **Blood Bank Request Component Test**

```

import React from 'react';

```

```
import { render, fireEvent, waitFor } from '@testing-library/react';

import axios from 'axios';

import BloodBankRequestComponent from
'../Component/BloodBankRequestComponent'; // Adjust the import path as necessary

jest.mock('axios');

describe('BloodBankRequestComponent', () => {

it('renders hospital data and can approve a request', async () => {

const hospitalData = [

{

'id': "10bab100-d6b3-41d0-9d3b-46491cda7ca0",

'name': "ABC Blood Bank",

'email': "sanjaihari2002@gmail.com",

'phoneNumber': 6382202487,

'location': "Chennai",

'governmentId': "1245433",

'document': "document",

'doorNo': "24",

'street': "VK Street,",

'area': "Ktc Nagar",

'city': "Viruthunagar",

'state': "Tamil Nadu",

'postalCode': "627011",

'status': 0
```

```

}

];

axios.get.mockResolvedValue({ data: hospitalData });

const { getByTestId } = render(<BloodBankRequestComponent />);

await waitFor(() => expect(getByTestId('row-10bab100-d6b3-41d0-9d3b-
46491cda7ca0')).toBeInTheDocument());

});

});

```

### **Add Blood Stock Component Test**

```

import React from 'react';

import { render, fireEvent, waitFor } from '@testing-library/react';

import AddBloodStockComponent from '../Component/AddBloodStockComponent';

import axios from 'axios';

import Cookies from 'js-cookie';

jest.mock('axios');

jest.mock('js-cookie', () => ({

  get: jest.fn(),

}));

describe('AddBloodStockComponent', () => {

  beforeEach(() => {

    jest.clearAllMocks();

  });

  it('renders correctly', () => {

```

```
const { getByTestId } = render(<AddBloodStockComponent />);

expect(getByTestId('headname')).toHaveTextContent('Add Blood Stock');

});

it('submits the form with correct blood type and units', async () => {

  Cookies.get.mockReturnValue('scbc238wej873cewd72');

  const { getByTestId } = render(<AddBloodStockComponent />);

  const bloodTypeSelect = getByTestId('BloodType');

  const unitInput = getByTestId('Unit');

  const addButton = getByTestId('addbutton');

  fireEvent.change(bloodTypeSelect, { target: { value: 'A+ve' } });

  fireEvent.change(unitInput, { target: { value: '5' } });

  axios.post.mockResolvedValue({ status: 200 });

  fireEvent.click(addButton);

  await waitFor(() => {

    expect(axios.post).toHaveBeenCalledWith('https://localhost:7089/api/BloodStock', {

      bloodStockId: '',

      bloodType: 'A+ve',

      units: 5,

      accountId: 'scbc238wej873cewd72',

    });

  });

});

});

});

});
```

## Add Blood Camp Test

```
import React from 'react';

import { render, fireEvent, waitFor } from '@testing-library/react';

import '@testing-library/jest-dom';

import AddBloodCampComponent from '../Component/AddBloodCampComponent';

import axios from 'axios';

import Cookies from 'js-cookie';

jest.mock('axios');

jest.mock('js-cookie');

describe('AddBloodCampComponent', () => {

  beforeEach(() => {

    Cookies.get.mockReturnValue('test-id');

  });

  it('renders without crashing', () => {

    const { getByTestId } = render(<AddBloodCampComponent />);

    expect(getByTestId('headname')).toHaveTextContent('Blood Camp');

  });

  it('submits the form with correct data', async () => {

    axios.post.mockResolvedValue({ status: 200 });

    const { getByTestId } = render(<AddBloodCampComponent />);

    fireEvent.change(getByTestId('bloodCampName'), { target: { value: 'Test Camp' } });

    fireEvent.change(getByTestId('bloodCampLocation'), { target: { value: 'Test Location' } });

    fireEvent.change(getByTestId('date'), { target: { value: '2024-01-30' } });
```

```
fireEvent.change(getByTestId('time'), { target: { value: '10:00' } });

fireEvent.click(getByTestId('bloodcampbtn'));

await waitFor(() => {

  expect(axios.post).toHaveBeenCalledWith('https://localhost:7089/api/BloodCamp', {

    bloodCampName: 'Test Camp',

    bloodCampLocation: 'Test Location',

    date: '2024-01-30',

    time: '10:00',

    accountId: 'test-id'

  });

});

});

});

});
```

## Output

```
✓ renders correctly (229 ms)
✓ submits the form with valid data (45 ms)

Test Suites: 13 passed, 13 total
Tests:       20 passed, 20 total
Snapshots:   0 total
Time:        25.688 s
Ran all test suites related to changed files.
```

```
(c) Microsoft Corporation. All rights reserved.
D:\MicroProject\bloodbankmanagementapp> npm test
> bloodbankmanagementapp@0.1.0 test
> react-scripts test --verbose
```

```
C:\Windows\system32\cmd.exe
at validatePropertiesInDevelopment (node_modules/react-dom/cjs/react-dom.development.js:9541:15)
at setInitialProperties (node_modules/react-dom/cjs/react-dom.development.js:9838:5)
at finalizeInitialChildren (node_modules/react-dom/cjs/react-dom.development.js:10958:1)
at completeWork (node_modules/react-dom/cjs/react-dom.development.js:22391:17)
at completeUnitOfWork (node_modules/react-dom/cjs/react-dom.development.js:26593:16)
at performUnitOfWork (node_modules/react-dom/cjs/react-dom.development.js:26568:5)
at workLoopSync (node_modules/react-dom/cjs/react-dom.development.js:26446:5)
at renderRootSync (node_modules/react-dom/cjs/react-dom.development.js:25738:74)
at performConcurrentWorkOnRoot (node_modules/react-dom/cjs/react-dom.development.js:25627:24)
at act (node_modules/react/cjs/react.development.js:2582:11)
at node_modules/@testing-library/react/dist/act-compat.js:84:25
at renderRoot (node_modules/@testing-library/react/dist/pure.js:159:26)
at render (node_modules/@testing-library/react/dist/pure.js:246:18)
at Object.<anonymous> (src/ChangePasswordTest.test.js:13:35)

console.log
  { data: { emailExists: true, sendMail: true } }
    at src/Component/ForgetPasswordComponent.js:37:17

PASS src/ForgetPasswordTest.test.js (16.809 s)
  ForgetPasswordComponent
    ✓ renders correctly (223 ms)
    ✓ submits the form with valid email (41 ms)

console.log
  { data: { emailExists: true, passcheck: true } }
    at src/Component/ChangePasswordComponent.js:54:17

PASS src/ChangePasswordTest.test.js (16.103 s)
  ChangePasswordComponent
    ✓ renders correctly (229 ms)
    ✓ submits the form with valid data (45 ms)

Test Suites: 13 passed, 13 total
Tests:       20 passed, 20 total
Snapshots:   0 total
Time:        25.688 s
Ran all test suites related to changed files.

Watch Usage
  > Press a to run all tests.
  > Press f to run only failed tests.
  > Press q to quit watch mode.
  > Press p to filter by a filename regex pattern.
  > Press t to filter by a test name regex pattern.
  > Press Enter to trigger a test run.
```