1. Form Submission Component

jsx

Copy code

```jsx
import React, { useState } from 'react';
import axios from 'axios';

const HealthRiskForm = () => {
  const [answers, setAnswers] = useState([]);
  const [userId, setUserId] = useState('');

  const handleAnswerChange = (questionId, score) => {
    setAnswers(prev => [...prev, { question: questionId, score }]);
  };

  const handleSubmit = async () => {
    try {
      const response = await axios.post('/api/forms/submit/', {
        user_id: userId,
        answers: answers,
      });
      // Handle PDF response
      const blob = new Blob([response.data], { type: 'application/pdf' });
      const url = window.URL.createObjectURL(blob);
      const link = document.createElement('a');
      link.href = url;
      link.setAttribute('download', 'report.pdf');
      document.body.appendChild(link);
      link.click();
      link.remove();
```

```
        } catch (error) {
            console.error("Error submitting form:", error);
        }
    };

    return (
        <div>
            <h1>Health Risk Assessment Form</h1>
            {/* Render your questions here */}
            <button onClick={handleSubmit}>Submit</button>
        </div>
    );
};

export default HealthRiskForm;
```

2. Dashboard Component

jsx

Copy code

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';

const Dashboard = () => {
    const [companyData, setCompanyData] = useState(null);

    useEffect(() => {
        const fetchData = async () => {
            try {
                const response = await axios.get('/api/user-stats/');
                setCompanyData(response.data);
```

```
      } catch (error) {
        console.error("Error fetching dashboard data:", error);
      }
    };

    fetchData();
  }, []);

  return (
    <div>
      <h1>Company Dashboard</h1>
      {companyData && (
        <div>
          <p>Total Users: {companyData.total_users}</p>
          <p>Total Questions: {companyData.total_questions}</p>
          <p>Total Reports: {companyData.total_reports}</p>
        </div>
      )}
    </div>
  );
};

export default Dashboard;
```

3. Report Generation Component

jsx

Copy code

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
```

```
const Report = ({ userId }) => {
  const [report, setReport] = useState(null);

  useEffect(() => {
    const fetchReport = async () => {
      try {
        const response = await axios.get(`/api/reports/${userId}/`);
        setReport(response.data);
      } catch (error) {
        console.error("Error fetching report:", error);
      }
    };

    fetchReport();
  }, [userId]);

  return (
    <div>
      <h1>Health Risk Report</h1>
      {report && (
        <div>
          <p>Name: {report.name}</p>
          <p>Health Risk Score: {report.health_risk_score}</p>
          {/* Add more report details here */}
        </div>
      )}
    </div>
  );
};
```

```
export default Report;
```

4. Collective Report Component

jsx

Copy code

```jsx
import React, { useEffect, useState } from 'react';
import axios from 'axios';

const CollectiveReport = () => {
  const [collectiveData, setCollectiveData] = useState(null);

  useEffect(() => {
    const fetchCollectiveReport = async () => {
      try {
        const response = await axios.get('/api/collective-report/?company=1'); // Replace with dynamic company ID
        setCollectiveData(response.data);
      } catch (error) {
        console.error("Error fetching collective report:", error);
      }
    };

    fetchCollectiveReport();
  }, []);

  return (
    <div>
      <h1>Collective Company Report</h1>
      {collectiveData && (
        <div>
          <p>Total Respondents: {collectiveData.total_respondents}</p>
          <p>Average Age: {collectiveData.avg_age}</p>
          {/* Add more collective report details here */}
        </div>
      )}
    </div>
  );
};

export default CollectiveReport;
```

5. User Profile Update Component

jsx

Copy code

```
import React, { useState } from 'react';
import axios from 'axios';

const UserProfile = () => {
  const [profile, setProfile] = useState({ name: '', age: '', gender: '' });

  const handleChange = (e) => {
    const { name, value } = e.target;
    setProfile(prev => ({ ...prev, [name]: value }));
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const response = await axios.put('/api/update-profile/', profile);
      alert('Profile updated successfully');
    } catch (error) {
      console.error("Error updating profile:", error);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="text" name="name" value={profile.name} onChange={handleChange} placeholder="Name" />
      <input type="number" name="age" value={profile.age} onChange={handleChange} placeholder="Age" />
      <select name="gender" value={profile.gender} onChange={handleChange}>
        <option value="">Select Gender</option>
        <option value="Male">Male</option>
        <option value="Female">Female</option>
        <option value="Other">Other</option>
      </select>
      <button type="submit">Update Profile</button>
    </form>
  );
};

export default UserProfile;
```

Integration Steps

1. **Axios Setup**: Make sure to set up axios in your React application for making API requests. You can set a base URL if required.
2. **Routing**: Integrate these components into your React Router for navigation between forms, dashboards, and reports.

3. **State Management**: If your application grows complex, consider using state management libraries like Redux or Context API.
4. **Styling**: Style your components as per your application's design requirements.
5. **Testing**: Make sure to test each component with mock data to ensure API integrations work correctly.

6. Form List Component for Admin

Ye component admin ko forms dekhne aur unhe manage karne ki suvidha deta hai.

jsx

Copy code

```
import React, { useEffect, useState } from 'react';

import axios from 'axios';


const FormList = () => {

  const [forms, setForms] = useState([]);


  useEffect(() => {

    const fetchForms = async () => {

      try {

        const response = await axios.get('/api/forms/');

        setForms(response.data);

      } catch (error) {

        console.error("Error fetching forms:", error);

      }

    };


    fetchForms();

  }, []);


  return (
    <div>
      <h1>Available Forms</h1>
      <ul>
        {forms.map(form => (
          <li key={form.id}>
            <h2>{form.title}</h2>
            <p>{form.description}</p>
            {/* Add buttons for actions like editing or deleting the form */}
          </li>
        ))}
      </ul>
    </div>
  );
};

export default FormList;
```

7. Question Management Component

Ye component admins ko questions add ya edit karne ki suvidha deta hai.

jsx

Copy code

```
import React, { useState } from 'react';
import axios from 'axios';

const QuestionManager = () => {
  const [question, setQuestion] = useState('');
  const [type, setType] = useState('closed');

  const handleAddQuestion = async () => {
    try {
            await axios.post('/api/questions/', { question, type });
            alert('Question added successfully!');
            setQuestion('');
            setType('closed');
        } catch (error) {
            console.error("Error adding question:", error);
        }
    };

    return (
        <div>
            <h1>Add New Question</h1>
            <input
                type="text"
                value={question}
                onChange={(e) => setQuestion(e.target.value)}
                placeholder="Enter question"
            />
            <select value={type} onChange={(e) => setType(e.target.value)}>
                <option value="closed">Closed-Ended</option>
                <option value="open">Open-Ended</option>
            </select>
            <button onClick={handleAddQuestion}>Add Question</button>
        </div>
    );
};

export default QuestionManager;
```

8. Individual Response View Component

Ye component individual responses dekhne ke liye hai.

jsx

Copy code

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';

const IndividualResponse = ({ userId }) => {
  const [response, setResponse] = useState(null);

  useEffect(() => {
    const fetchResponse = async () => {
      try {
        const res = await axios.get(`/api/responses/${userId}/`);
        setResponse(res.data);
      } catch (error) {
        console.error("Error fetching response:", error);
      }
    };

    fetchResponse();
  }, [userId]);

  return (
    <div>
      <h1>Individual Response</h1>
      {response && (
        <div>
          <p>Name: {response.name}</p>
          <p>Health Risk Score: {response.health_risk_score}</p>
```

```
                    {/* Add more details as needed */}
                </div>
            )}
        </div>
    );
};

export default IndividualResponse;
```

9. Download Reports Component

Ye component reports download karne ki suvidha deta hai.

jsx

Copy code

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

const DownloadReports = () => {
    const handleDownload = async (reportId) => {
        try {
            const response = await axios.get(` /api/reports/download/${reportId}/` , { responseType: 'blob' });
            const blob = new Blob([response.data], { type: 'application/pdf' });
            const url = window.URL.createObjectURL(blob);
            const link = document.createElement('a');
            link.href = url;
            link.setAttribute('download', ` report_${reportId}.pdf` );
            document.body.appendChild(link);
            link.click();
            link.remove();
        } catch (error) {
```

```
      console.error("Error downloading report:", error);
    }
  };

  return (
    <div>
      <h1>Download Reports</h1>
      {/* Assume reportIds is an array of report IDs */}
      {reportIds.map(id => (
        <button key={id} onClick={() => handleDownload(id)}>Download Report {id}</button>
      ))}
    </div>
  );
};

export default DownloadReports;
```

10. Excel Export Component

Ye component company-wise responses ko Excel format mein export karne ki suvidha deta hai.

jsx

Copy code

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

const ExcelExport = () => {
  const handleExport = async () => {
    try {
      const response = await axios.get('/api/export/excel/', { responseType: 'blob' });
      const blob = new Blob([response.data], { type: 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet' });
      const url = window.URL.createObjectURL(blob);
      const link = document.createElement('a');
      link.href = url;
      link.setAttribute('download', 'company_responses.xlsx');
      document.body.appendChild(link);
      link.click();
      link.remove();
    } catch (error) {
      console.error("Error exporting to Excel:", error);
    }
  };

  return (
    <div>
      <h1>Export to Excel</h1>
      <button onClick={handleExport}>Download Company Responses</button>
    </div>
  );
};

export default ExcelExport;
```

Integration Steps

1. **API Endpoints**: Ensure that your Django backend has the required API endpoints to support these components.
2. **Routing**: Integrate these components into your application's routing structure.
3. **State Management**: If needed, consider using a state management library for better data handling.
4. **Testing**: Test each component to ensure they function correctly with the backend.