

UCS2601

INTERNET PROGRAMMING

MINI PROJECT

Submitted By

Dilsha Singh D (3122 22 5001 028)

Kushal Varma Guntimadugu (3122 22 5001 031)

Hannah S (3122 22 5001 032)



Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam – 603110

TABLE OF CONTENTS:

1. Aim
2. Requirements
3. Flow Diagram
4. Functionalities With Explanations
5. Code
6. Result With Screenshots
7. Learning outcomes

MOU DATA ACQUISITION AND MANAGEMENT

AIM

To develop a user-friendly web application for automated MOU data management, enabling secure entry, retrieval, tracking, and notification of MOU information using Excel-based storage without the use of a traditional database system.

SOFTWARE AND FUNCTIONAL REQUIREMENTS:

SOFTWARE REQUIREMENTS:

1. Frontend: HTML, CSS, ReactJS
2. Backend: Node.js (Express.js), Appscript
3. Storage: Google Sheets API for storing and retrieving data
4. Authentication and Registration: AppScript, Node.js
5. Notifications: Email, Appscript for email notifications

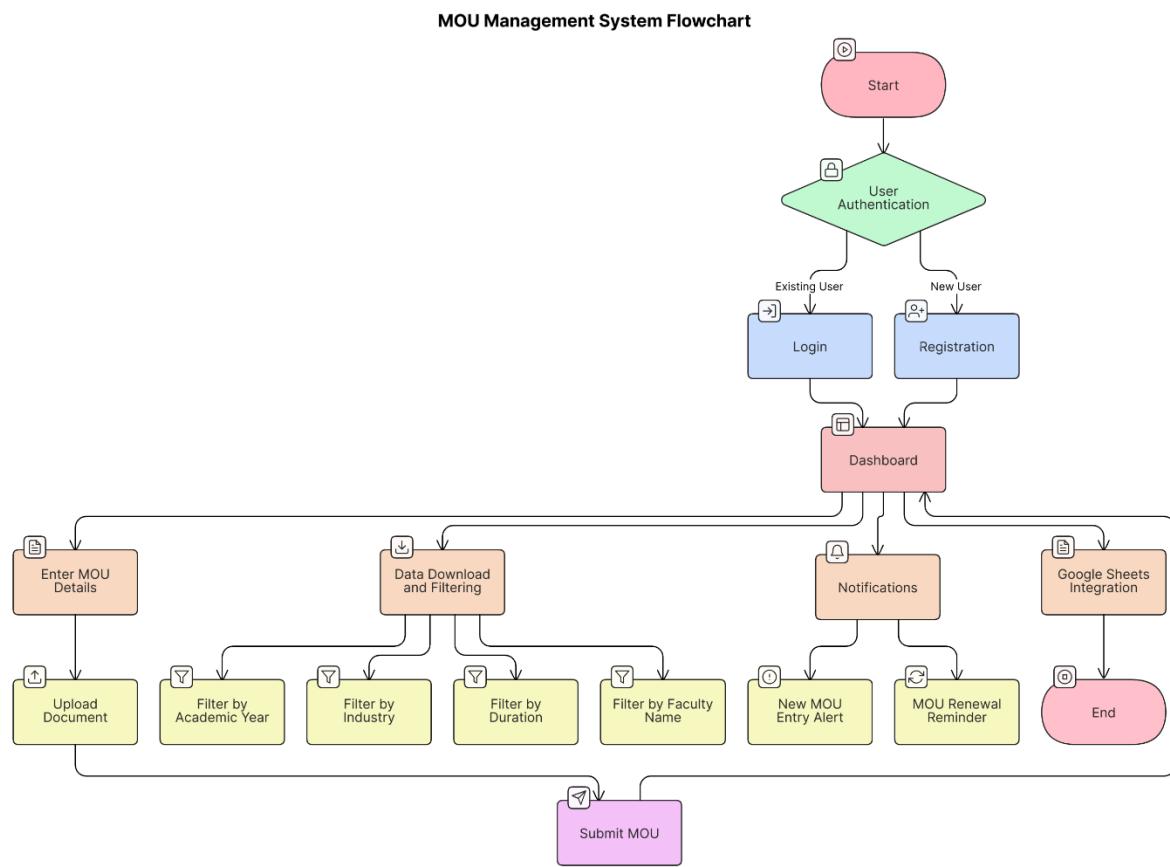
FUNCTIONAL REQUIREMENTS:

1. **User Authentication:** A secure login system developed using Node.js and Google Apps Script allows users to log in using their institutional email. New users register via a form, and access is role-based – faculty members can submit MOU data, while administrators can view and download records.
2. **MOU Data Submission:** Upon successful login, users can enter new MOU details such as Industry/Institute Name, Faculty Name, Faculty Involved, Duration, Purpose, Academic Year, and Outcomes. Users can also upload signed agreement documents. All data is stored and

dynamically updated in an Excel sheet (Google Sheet) using Apps Script.

3. **Filtered Data Retrieval and Excel-Based Downloads:** The system enables users to retrieve and filter MOU records based on Academic Year, Faculty Name, Industry/Institute, and MOU Duration. Filtered data is automatically updated in a central Google Sheet and can be downloaded directly by the admin for reporting.
4. **Email-Based Notifications for MOU Activities:** Email notifications are triggered using Apps Script to remind users about new MOU data entry and upcoming MOU renewals, based on duration. These alerts ensure users remain updated without relying on third-party services like Nodemailer.
5. **Interactive Frontend Interface:** Built using HTML, CSS, and ReactJS, the frontend provides a user-friendly interface for registration, login, and MOU data entry. It ensures smooth navigation, real-time form submission, and clear presentation of data filters and download options.

FLOW DIAGRAM



EXPLANATION OF FUNCTIONALITIES

1. Backend & Data Storage Setup

The backend is built using **Node.js (Express.js)** and **Google Apps Script**, which work together to handle data flow and backend logic.

MOU information is stored directly in **Google Sheets** using the **Google Sheets API**, allowing real-time read/write operations without relying on a traditional database.

2. User Login

Users log in through a **ReactJS-based frontend**, and credentials are authenticated using a backend developed in **Node.js and Apps Script**.

- Only verified users can log in and access the dashboard.
- On successful login, users are redirected to the main interface where they can submit and view MOU-related data. The system does not use roles; all authenticated users have equal access to the system functionalities.

3. MOU Data Entry

After logging in, users can fill out a form to submit MOU-related details. The fields include:

- Industry/Institute Name
- Faculty Name (Signatory) and Faculty Details Involved
- MOU Duration and Academic Year
- Purpose and Outcomes of the MOU
- Upload of the Signed Agreement Document

Upon submission, the data is written directly into the connected Google Sheet via **Apps Script**, ensuring it's immediately available for retrieval and download.

4. Filtered Data Retrieval and Download

Users can retrieve MOU records using filters such as:

- Academic Year
- Faculty Name
- Duration of MOU
- Industry/Institute

Filtered data is fetched in real-time from **Google Sheets**, displayed on the frontend, and available for download in **Excel format**.

5. Email Notification System

A **monthly email alert system** is integrated using **Google Apps Script**.

- Users receive reminders to enter new MOU records.
 - The system also sends reminders for MOU renewal based on each entry's duration.
- These automated alerts help keep the data up to date and ensure no MOU entries or renewals are missed.

CODE

src/components/auth/Login.jsx

```
import React, { useState } from 'react';
import './style.css';
import { useNavigate } from 'react-router-dom';
// Use a public CORS proxy URL for development.
const PROXY_URL = 'http://localhost:3000/proxy?url=';
const SCRIPT_URL = PROXY_URL +
encodeURIComponent('https://script.google.com/macros/s/AKfycby9Jtwc3g0uL2JEPMSB_ODJVH62f7hf0Ux9u
wjCiIf_epVhKGPUR6pDL-MkKh1UfdvkVw/exec');

const Login = ({ switchToRegister, onLoginSuccess }) => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');
  const [success, setSuccess] = useState('');
  const navigate = useNavigate();
  const handleLogin = async (e) => {
    e.preventDefault();
    setError('');
    setSuccess('');

    try {
      const response = await fetch(SCRIPT_URL, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify({
          action: 'login',
          email,
          password
        })
      });

      console.log('Raw response:', response); // Log entire response object
      const res = await response.json();
    }
  }
}
```

```

        console.log('Parsed response:', res); // Log parsed JSON response

        if (res.success) {
            setSuccess(res.message || 'Login successful!');
            localStorage.setItem('userEmail', email);
            onLoginSuccess();
        } else {
            setError(res.message || 'Invalid credentials');
        }
    } catch (err) {
        console.error('Error:', err);
        setError('Server error. Please try again later.');
    }
};

return (
    <div className="auth-container">
        <div className="auth-card">
            <h2 className="auth-header">Login</h2>
            {error && <p style={{ color: 'red', textAlign: 'center' }}>{error}</p>}
            {success && <p style={{ color: 'green', textAlign: 'center' }}>{success}</p>}
            <form onSubmit={handleLogin}>
                <div className="form-group">
                    <label>Email</label>
                    <input
                        type="email"
                        className="form-control"
                        value={email}
                        onChange={(e) => setEmail(e.target.value)}
                        required
                    />
                </div>
                <div className="form-group">
                    <label>Password</label>
                    <input
                        type="password"
                        className="form-control"
                        value={password}
                        onChange={(e) => setPassword(e.target.value)}
                        required
                    />
                </div>
                <button type="submit" className="btn-primary">Login</button>
            </form>
            <button className="btn-link" onClick={() => navigate('/Register')}>
                New user? Register
            </button>
        </div>
    </div>
);
};

export default Login;

```

src/components/auth/Register.jsx

```

import React, { useState } from 'react';
import './style.css';
const PROXY_URL = 'http://localhost:3000/proxy?url=';
const SCRIPT_URL = PROXY_URL +
encodeURIComponent('https://script.google.com/macros/s/AKfycby9Jtwc3g0uL2JEPMSB_ODJVH62f7hf0Ux9u
wjCiIf_epVhKGPU6pDL-MkKh1UfdvkVw/exec');
import { useNavigate } from 'react-router-dom';

```

```

const Register = ({ switchToLogin }) => {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [confirmPassword, setConfirmPassword] = useState('');
  const [error, setError] = useState('');
  const [success, setSuccess] = useState('');
  const navigate = useNavigate();

  const handleRegistration = async (e) => {
    e.preventDefault();
    setError('');
    setSuccess('');

    if (password !== confirmPassword) {
      setError('Passwords do not match');
      return;
    }

    try {
      const response = await fetch(SCRIPT_URL, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify({
          action: 'register',
          name: name.trim(),
          email: email.trim().toLowerCase(),
          password: password.trim()
        })
      });

      console.log('Raw response:', response); // Log entire response object

      const res = await response.json();
      console.log('Parsed response:', res); // Log parsed JSON response

      if (res.success) {
        setSuccess(res.message || 'Registered successfully!');
        setName('');
        setEmail('');
        setPassword('');
        setConfirmPassword('');
      } else {
        setError(res.message || 'Registration failed');
      }
    } catch (err) {
      console.error('Error:', err);
      setError('Something went wrong. Please try again later.');
    }
  };
}

return (
  <div className="auth-container">
    <div className="auth-card">
      <h2 className="auth-header">Register</h2>
      {error && <p style={{ color: 'red', textAlign: 'center' }}>{error}</p>}
      {success && <p style={{ color: 'green', textAlign: 'center' }}>{success}</p>}
      <form onSubmit={handleRegistration}>
        <div className="form-group">
          <label>Name</label>
          <input
            type="text"
            className="form-control"
            value={name}
            onChange={(e) => setName(e.target.value)}
            required
          />
        </div>
      </form>
    </div>
  </div>
)

```

```

        <div className="form-group">
          <label>Email</label>
          <input
            type="email"
            className="form-control"
            value={email}
            onChange={(e) => setEmail(e.target.value)}
            required
          />
        </div>
        <div className="form-group">
          <label>Password</label>
          <input
            type="password"
            className="form-control"
            value={password}
            onChange={(e) => setPassword(e.target.value)}
            required
          />
        </div>
        <div className="form-group">
          <label>Confirm Password</label>
          <input
            type="password"
            className="form-control"
            value={confirmPassword}
            onChange={(e) => setConfirmPassword(e.target.value)}
            required
          />
        </div>
        <button type="submit" className="btn-primary">Register</button>
      </form>
      <button className="btn-link" onClick={() => navigate('/login')}>
        Already have an account?
      </button>
    </div>
  );
};

export default Register;

```

src/components/auth/style.css

```

@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap');

:root {
  --primary: #4a90e2;
  --primary-dark: #357ab8;
  --bg-light: #f4f7fc;
  --text-dark: #2c3e50;
  --card-bg: #ffffff;
  --input-bg: #ecf0f1;
}

html, body {
  height: 100%;
  margin: 0;
  padding: 0;
  font-family: 'Roboto', sans-serif;
  background: var(--bg-light);
}

.auth-container {
  height: 100vh;
  padding: 2rem 0;
}

```

```
    box-sizing: border-box;
    display: flex;
    justify-content: center;
    align-items: center;
}

.auth-card {
  background: var(--card-bg);
  padding: 2.5rem;
  border-radius: 1rem;
  box-shadow: 0 8px 16px rgba(0,0,0,0.1);
  width: 380px;
  max-width: 90%;
}

.auth-header {
  text-align: center;
  margin-bottom: 2rem;
  font-size: 1.75rem;
  color: var(--text-dark);
  font-weight: 500;
}

.form-group {
  margin-bottom: 1.25rem;
}

.form-group label {
  display: block;
  margin-bottom: 0.5rem;
  font-weight: 500;
  color: var(--text-dark);
}

.form-control {
  width: 100%;
  padding: 0.75rem 1rem;
  background: var(--input-bg);
  border: none;
  border-radius: 0.5rem;
  font-size: 1rem;
  color: var(--text-dark);
  transition: background 0.3s;
}

.form-control:focus {
  background: #fff;
  outline: 2px solid var(--primary);
}

.btn-primary {
  width: 100%;
  padding: 0.75rem;
  background: var(--primary);
  color: #fff;
  border: none;
  border-radius: 0.5rem;
  cursor: pointer;
  font-size: 1rem;
  font-weight: 500;
  transition: background 0.3s;
}

.btn-primary:hover {
  background: var(--primary-dark);
}

.btn-link {
  display: block;
  width: 100%;
```

```

margin-top: 1rem;
background: none;
border: none;
color: var(--primary);
cursor: pointer;
text-align: center;
text-decoration: none;
font-size: 0.9rem;
}

.btn-link:hover {
  text-decoration: underline;
}

```

src/components/common/Footer.jsx

```

import React from 'react';
import { Link } from 'react-router-dom';

const Footer = () => {
  return (
    <footer className="footer">
      <div className="container">
        <div className="footer-content">
          <div className="footer-section">
            <h3>Quick Links</h3>
            <ul>
              <li><Link to="/">Home</Link></li>
              <li><Link to="/about">About</Link></li>
              <li><Link to="/faq">FAQ</Link></li>
            </ul>
          </div>
          <div className="footer-section">
            <h3>Contact</h3>
            <p>Email: contact@moutracker.edu</p>
            <p>Phone: +1 (123) 456-7890</p>
          </div>
        </div>
        <div className="copyright">
          <p>© {new Date().getFullYear()} MOUTracker. All rights reserved.</p>
        </div>
      </div>
    </footer>
  );
}

export default Footer;

```

src/components/common/Header.jsx

```

import React from 'react';
import { Link } from 'react-router-dom';

const Header = ({ isLoggedIn }) => {
  return (
    <header className="app-header">
      <div className="logo">
        <Link to="/">MOUTracker</Link>
      </div>
      <nav>
        <ul>

```

```

        <li><Link to="/about">About</Link></li>
        <li><Link to="/faq">FAQ</Link></li>
        <li><Link to="/contact">Contact</Link></li>
        <li><Link to="/newsroom">Newsroom</Link></li>
        {isLoggedIn ? (
            <li><Link to="/dashboard">Dashboard</Link></li>
        ) : (
            <>
                <li><Link to="/login">Login</Link></li>
                <li><Link to="/register">Register</Link></li>
            </>
        )}
    </ul>
</nav>
</header>
);
};

export default Header;

```

src/components/Dashboard/dashboard.jsx

```

import React from 'react';
import { Link } from 'react-router-dom';

const Dashboard = ({ onLogout }) => {
    return (
        <div className="dashboard">
            <header className="dashboard-header">
                <h1>Dashboard</h1>
                <button onClick={onLogout} className="btn btn-logout">Logout</button>
            </header>

            <div className="dashboard-grid">
                <Link to="/mou-form" className="dashboard-card">
                    <h2>Add New MOU</h2>
                    <p>Enter details of a new Memorandum of Understanding</p>
                </Link>

                <Link to="/mou-download" className="dashboard-card">
                    <h2>Download MOU Data</h2>
                    <p>Filter and download existing MOU information</p>
                </Link>

                <Link to="/notifications" className="dashboard-card">
                    <h2>Notifications</h2>
                    <p>View alerts and reminders</p>
                </Link>

                <Link to="/services/monthly-report" className="dashboard-card">
                    <h2>Monthly Report</h2>
                    <p>Generate and send monthly MOU reports</p>
                </Link>
            </div>
        </div>
    );
};

export default Dashboard;

```

src/components/Dashboard/MonthlyReport.jsx

```
import React, { useState } from 'react';
import { sendMonthlyReportEmail } from '../services/emailService';

const MonthlyReport = () => {
  const [email, setEmail] = useState('');
  const [isSending, setIsSending] = useState(false);
  const [message, setMessage] = useState('');

  const handleSubmit = async (e) => {
    e.preventDefault();
    setIsSending(true);
    setMessage('');

    try {
      const result = await sendMonthlyReportEmail(email);
      setMessage(result.message);
    } catch (error) {
      setMessage('Error sending report: ' + error.message);
    } finally {
      setIsSending(false);
    }
  };

  return (
    <div className="monthly-report">
      <h2>Monthly MOU Report</h2>
      <form onSubmit={handleSubmit}>
        <div className="form-group">
          <label htmlFor="email">Recipient Email:</label>
          <input
            type="email"
            id="email"
            value={email}
            onChange={(e) => setEmail(e.target.value)}
            required
          />
        </div>
        <button type="submit" disabled={isSending}>
          {isSending ? 'Sending...' : 'Send Monthly Report'}
        </button>
      </form>
      {message && <div className="message">{message}</div>}
    </div>
  );
};

export default MonthlyReport;
```

src/components/Dashboard/MOUDownload.jsx

```
import React, { useState, useEffect } from 'react';
import './MOUFilterDownload.css';

const MOUFilterDownload = () => {
  const [data, setData] = useState([]);
  const [filteredData, setFilteredData] = useState([]);
  const [filters, setFilters] = useState({
    academicYear: '',
    industry: '',
    duration: '',
    faculty: ''
  });
};
```

```

useEffect(() => {
  fetch('https://script.google.com/macros/s/AKfycbzRife_Q5P91CIhJ4j-tRqqgx4SD9jsJIOcz6sMxdIZfmFKbrlquQvCNrqIKw5yi4Y7w/exec')
    .then(res => res.json())
    .then(result => {
      setData(result.data || []);
      setFilteredData(result.data || []);
    })
    .catch(err => console.error('Fetch error:', err));
}, []);

const handleChange = (e) => {
  const { name, value } = e.target;
  const newFilters = { ...filters, [name]: value };
  setFilters(newFilters);

  const filtered = data.filter(entry => {
    const entryDuration = getDuration(entry['Start Date'], entry['End Date']);
    return (
      (newFilters.academicYear === '' || entry['Academic Year'] === newFilters.academicYear) &&
      (newFilters.industry === '' || (entry['Industry/Institute Name'] || '').toLowerCase().includes(newFilters.industry.toLowerCase())) &&
      (newFilters.faculty === '' || (entry['Faculty Name'] || '').toLowerCase().includes(newFilters.faculty.toLowerCase())) &&
      (newFilters.duration === '' || entryDuration === newFilters.duration)
    );
  });
  setFilteredData(filtered);
};

const getDuration = (start, end) => {
  if (!start || !end) return '';
  const s = new Date(start);
  const e = new Date(end);
  if (isNaN(s) || isNaN(e)) return '';
  const diff = e.getFullYear() - s.getFullYear();
  return `${diff} year${diff !== 1 ? 's' : ''}`;
};

const downloadCSV = () => {
  if (filteredData.length === 0) {
    alert('No data to download!');
    return;
  }

  const headers = Object.keys(filteredData[0]);
  const rows = filteredData.map(row =>
    headers.map(field => `${row[field]} || ''`).join(',')
  );
  const csv = [headers.join(','), ...rows].join('\n');

  const blob = new Blob([csv], { type: 'text/csv' });
  const url = URL.createObjectURL(blob);
  const link = document.createElement('a');
  link.href = url;
  link.download = 'filtered_mou_data.csv';
  link.click();
};

return (
  <div className="filter-download-container">
    <h3> Filter & Download MOU Data </h3>
    <div className="filters">
      <label>
        Academic Year:

```

```

        <select name="academicYear" onChange={handleChange}>
          <option value="">All</option>
          <option value="2022-2023">2022-2023</option>
          <option value="2023-2024">2023-2024</option>
          <option value="2024-2025">2024-2025</option>
          <option value="2025-2026">2025-2026</option>
        </select>
      </label>

      <label>
        Industry/Institute:
        <input name="industry" type="text" onChange={handleChange} placeholder="Search..." />
      </label>

      <label>
        Faculty Name:
        <input name="faculty" type="text" onChange={handleChange} placeholder="Search..." />
      </label>

      <label>
        Duration:
        <select name="duration" onChange={handleChange}>
          <option value="">All</option>
          <option value="0 years">0 years</option>
          <option value="1 year">1 year</option>
          <option value="2 years">2 years</option>
          <option value="3 years">3 years</option>
          <option value="4 years">4 years</option>
        </select>
      </label>
    </div>

    <button onClick={downloadCSV}>⬇️ Download CSV</button>
    <p>{filteredData.length} entries found.</p>
  </div>
};

export default MOUFilterDownload;

```

src/components/Dashboard/MOUFilterDownload.css

```

/* Global styles */
body {
  font-family: Arial, sans-serif;
  background-color: #f4f7fc;
  color: #333;
  margin: 0;
  padding: 0;
}

/* Container for filter and download section */
.filter-download-container {
  max-width: 800px;
  margin: 40px auto;
  padding: 20px;
  background-color: #ffffff;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
}

h3 {
  text-align: center;
  font-size: 24px;
  color: #333;
}

```

```
    margin-bottom: 20px;
}

.filters {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
  gap: 15px;
  margin-bottom: 30px;
}

/* Label and input styles */
label {
  font-weight: 600;
  font-size: 14px;
  color: #555;
  display: block;
  margin-bottom: 8px;
}

input, select {
  width: 100%;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 14px;
  box-sizing: border-box;
  background-color: #f9f9f9;
}

input:focus, select:focus {
  outline: none;
  border-color: #007BFF;
  background-color: #fff;
}

/* Button styles */
button {
  display: block;
  width: 100%;
  padding: 12px;
  background-color: #007BFF;
  color: white;
  font-size: 16px;
  font-weight: bold;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #0056b3;
}

button:active {
  background-color: #004085;
}

button:disabled {
  background-color: #cccccc;
  cursor: not-allowed;
}

/* Info text */
p {
  text-align: center;
  font-size: 14px;
  color: #777;
  margin-top: 10px;
}
```

```

/* Responsive Design */
@media (max-width: 768px) {
  .filter-download-container {
    padding: 15px;
    margin: 20px;
  }

  .filters {
    grid-template-columns: 1fr;
  }

  button {
    width: 100%;
    padding: 15px;
    font-size: 18px;
  }
}

```

src/components/Dashboard/MOUForm.jsx

```

import React, { useState, useEffect } from 'react';
import './moustyle.css';

const MOUForm = () => {
  const [userEmail, setUserEmail] = useState('');

  const [formData, setFormData] = useState({
    'Industry/Institute Name': '',
    'Start Date': '',
    'End Date': '',
    'Faculty Name': '',
    'Faculty Details': '',
    'Academic Year': '',
    'Purpose of MOU': '',
    'Outcomes of MOU': '',
    'User Email': ''
  });
  useEffect(() => {
    const email = localStorage.getItem('userEmail');
    if (email) {
      setUserEmail(email);
      setFormData(prev => ({
        ...prev,
        'User Email': email
      }));
    }
  }, []);

  const [fileData, setFileData] = useState(null);
  const [status, setStatus] = useState('');

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

  const handleFileChange = (e) => {
    const uploadedFile = e.target.files[0];
    const allowedTypes = ['application/pdf', 'application/msword',
    'application/vnd.openxmlformats-officedocument.wordprocessingml.document', 'image/jpeg',
    'image/png'];
  };
}

```

```

    if (uploadedFile && allowedTypes.includes(uploadedFile.type)) {
      const reader = new FileReader();
      reader.onloadend = () => {
        const base64 = reader.result.split(',')[1];
        setFileData({
          base64,
          name: uploadedFile.name,
          type: uploadedFile.type
        });
      };
      reader.readAsDataURL(uploadedFile);
    } else {
      alert('Invalid file type. Please upload a PDF, DOC, DOCX, PNG, or JPG file.');
    }
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    setStatus('Submitting...');

    const submissionData = {
      ...formData,
      'User Email': userEmail // Ensure email is included
    };

    const form = new FormData();
    Object.keys(submissionData).forEach((key) => {
      form.append(key, submissionData[key]);
    });

    if (fileData) {
      form.append('file', JSON.stringify(fileData));
    }

    try {
      const response = await fetch('https://script.google.com/macros/s/AKfycbzRifE_Q5P91CIhJ4j-tRqqgx4SD9j-sJI0cz6sMxdIZfmFKbrlquQvCNrqIKw5yi4Y7w/exec', {
        method: 'POST',
        body: form
      });

      const result = await response.json();

      if (result.result === 'success') {
        setStatus('✅ MOU submitted successfully!');
        setFormData({
          'Industry/Institute Name': '',
          'Start Date': '',
          'End Date': '',
          'Faculty Name': '',
          'Faculty Details': '',
          'Academic Year': '',
          'Purpose of MOU': '',
          'Outcomes of MOU': ''
        });
        setFileData(null);
      } else {
        setStatus('✖ Error: ' + result.error);
      }
    } catch (err) {
      setStatus('✖ Network error. Try again later.');
      console.error(err);
    }
  };
};

return (
  <div className="form-container">
    <h2>MOU Submission Form</h2>

```

```

    {status && <p>{status}</p>}
    <form onSubmit={handleSubmit} encType="multipart/form-data">
      <label>Industry/Institute Name</label>
      <input type="text" name="Industry/Institute Name" value={formData['Industry/Institute Name']} onChange={handleChange} required />

      <label>Start Date of MOU</label>
      <input type="date" name="Start Date" value={formData['Start Date']} onChange={handleChange} required />

      <label>End Date of MOU</label>
      <input type="date" name="End Date" value={formData['End Date']} onChange={handleChange} required />

      <label>Upload Signed Agreement Document</label>
      <input type="file" accept=".pdf,.doc,.docx,.png,.jpg" onChange={handleFileChange} required />

      <label>Faculty Name</label>
      <input type="text" name="Faculty Name" value={formData['Faculty Name']} onChange={handleChange} required />

      <label>Faculty Details</label>
      <input type="text" name="Faculty Details" value={formData['Faculty Details']} onChange={handleChange} required />

      <label>Academic Year</label>
      <select name="Academic Year" value={formData['Academic Year']} onChange={handleChange} required>
        <option value="">Select Academic Year</option>
        <option value="2022-2023">2022-2023</option>
        <option value="2023-2024">2023-2024</option>
        <option value="2024-2025">2024-2025</option>
        <option value="2025-2026">2025-2026</option>
      </select>

      <label>Purpose of MOU</label>
      <input type="text" name="Purpose of MOU" value={formData['Purpose of MOU']} onChange={handleChange} required />

      <label>Outcomes of MOU</label>
      <textarea name="Outcomes of MOU" value={formData['Outcomes of MOU']} onChange={handleChange} required />

      <button type="submit">Submit MOU</button>
    </form>
  </div>
};

};

export default MOUForm;

```

src/components/Dashboard/moustyle.css

```

/* moustyle.css */

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background: #f4f6f9;
  margin: 0;
  padding: 0;
}

.form-container {
  max-width: 800px;

```

```
    margin: 50px auto;
    padding: 40px 50px;
    background-color: #ffffff;
    border-radius: 12px;
    box-shadow: 0 6px 20px rgba(0, 0, 0, 0.1);
}

h2 {
    text-align: center;
    color: #2c3e50;
    margin-bottom: 30px;
    font-size: 28px;
}

form {
    display: flex;
    flex-direction: column;
}

label {
    font-weight: 600;
    margin-top: 20px;
    margin-bottom: 8px;
    color: #34495e;
    font-size: 16px;
}

input[type="text"],
input[type="date"],
input[type="file"],
select,
textarea {
    padding: 12px 15px;
    font-size: 16px;
    border: 1px solid #d0d7de;
    border-radius: 6px;
    background-color: #fefefe;
    color: #333; /* Ensure text color is dark for better visibility */
    transition: border 0.3s ease;
}

input[type="text"]:focus,
input[type="date"]:focus,
select:focus,
textarea:focus {
    border-color: #3498db;
    outline: none;
    background-color: #ffffff;
}

textarea {
    resize: vertical;
    min-height: 100px;
}

button {
    margin-top: 30px;
    padding: 14px 0;
    font-size: 16px;
    font-weight: 600;
    color: white;
    background-color: #3498db;
    border: none;
    border-radius: 8px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #2c80b4;
```

```

}

p {
  text-align: center;
  font-weight: 500;
  margin-bottom: 20px;
  color: #27ae60;
  font-size: 16px;
}

input[type="date"] {
  color: #333; /* Make the date input text color more visible */
}

input[type="file"] {
  background-color: #e0e0e0; /* Make the file input background more visible */
}

@media (max-width: 600px) {
  .form-container {
    padding: 20px;
  }

  h2 {
    font-size: 24px;
  }

  label,
  input,
  select,
  textarea,
  button {
    font-size: 15px;
  }
}

```

src/components/Dashboard/Notifications.css

```

.notifications-container {
  max-width: 800px;
  margin: 2rem auto;
  padding: 2rem;
  background: white;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.notification-prefs {
  margin: 2rem 0;
}

.pref-item {
  padding: 1rem;
  border-bottom: 1px solid #eee;
}

.pref-item label {
  display: flex;
  align-items: center;
  gap: 1rem;
  font-weight: 500;
  cursor: pointer;
}

```

```

.pref-item input[type="checkbox"] {
  width: 18px;
  height: 18px;
}

.pref-description {
  margin: 0.5rem 0 2rem;
  color: #666;
  font-size: 0.9rem;
}

.email-display {
  background: #f8f9fa;
  padding: 1rem;
  border-radius: 4px;
  margin: 1.5rem 0;
}

.save-btn {
  background: #3498db;
  color: white;
  border: none;
  padding: 0.75rem 1.5rem;
  border-radius: 4px;
  cursor: pointer;
  font-size: 1rem;
}

.save-btn:hover {
  background: #2980b9;
}

.save-btn:disabled {
  background: #95a5a6;
  cursor: not-allowed;
}

.status-message {
  margin-top: 1rem;
  padding: 0.5rem;
  border-radius: 4px;
}

.status-message.success {
  background: #e8f8f0;
  color: #2ecc71;
}

.status-message.error {
  background: #fdecea;
  color: #e74c3c;
}

```

src/components/Dashboard/Notifications.jsx

```

import React, { useState, useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import './Notifications.css';

const Notifications = () => {
  const [notificationPrefs, setNotificationPrefs] = useState({
    newMOU: true,
    renewalReminders: true,
    monthlySummary: true
  });

```

```
const [email, setEmail] = useState('');
const [isLoading, setIsLoading] = useState(false);
const [status, setStatus] = useState('');

useEffect(() => {
  // Fetch user's email from localStorage or context
  const userEmail = localStorage.getItem('userEmail') || '';
  setEmail(userEmail);
}, []);

const handlePrefChange = (e) => {
  const { name, checked } = e.target;
  setNotificationPrefs(prev => ({
    ...prev,
    [name]: checked
  }));
};

const savePreferences = async () => {
  setIsLoading(true);
  setStatus('');

  try {
    const response = await fetch('https://script.google.com/macros/s/AKfycbzRifE_Q5P91CIhJ4jtRqqgx4SD9j-sJI0cz6sMxdIZfmFKbrlquQvCNrqIKw5yi4Y7w/exec', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        action: 'updateNotificationPrefs',
        email,
        preferences: notificationPrefs
      })
    });

    const result = await response.json();
    if (result.success) {
      setStatus('Preferences saved successfully!');
    } else {
      setStatus('Error saving preferences');
    }
  } catch (error) {
    setStatus('Network error. Please try again.');
  } finally {
    setIsLoading(false);
  }
};

return (
  <div className="notifications-container">
    <h2>Email Notification Preferences</h2>
    <p>Manage how you receive notifications about MOUs</p>

    <div className="notification-prefs">
      <div className="pref-item">
        <label>
          <input
            type="checkbox"
            name="newMOU"
            checked={notificationPrefs.newMOU}
            onChange={handlePrefChange}
          />
          New MOU Entries
        </label>
        <p className="pref-description">Receive emails when new MOUs are added</p>
      </div>

      <div className="pref-item">
        <label>
```

```

        <input
          type="checkbox"
          name="renewalReminders"
          checked={notificationPrefs.renewalReminders}
          onChange={handlePrefChange}
        />
      Renewal Reminders
    </label>
    <p className="pref-description">Get reminders before MOUs expire</p>
  </div>

  <div className="pref-item">
    <label>
      <input
        type="checkbox"
        name="monthlySummary"
        checked={notificationPrefs.monthlySummary}
        onChange={handlePrefChange}
      />
      Monthly Summary
    </label>
    <p className="pref-description">Receive a monthly report of all MOUs</p>
  </div>
</div>

<div className="email-display">
  <p>Notifications will be sent to: <strong>{email}</strong></p>
</div>

<button
  onClick={savePreferences}
  disabled={isLoading}
  className="save-btn"
>
  {isLoading ? 'Saving...' : 'Save Preferences'}
</button>

{status && <p className="status-message">{status}</p>}
</div>
);
};

export default Notifications;

```

src/pages/About.jsx

```

import React from 'react';
import Header from '../common/Header';
import Footer from '../common/Footer';

const About = () => {
  return (
    <div className="page">
      <Header />

      <main className="main-content">
        <div className="container">
          <h1>About MOUTracker</h1>
          <section>
            <h2>Our Mission</h2>
            <p>
              MOUTracker was developed to streamline the management of Memorandums of Understanding between educational institutions and industry partners. Our goal is to automate the
            </p>
          </section>
        </div>
      </main>
    </div>
  );
};

export default About;

```

```

        tracking process and ensure no important deadlines are missed.
    </p>
</section>

<section>
    <h2>Features</h2>
    <ul>
        <li>Centralized MOU data management</li>
        <li>Automated renewal reminders</li>
        <li>Easy data entry and retrieval</li>
        <li>Customizable filtering options</li>
        <li>Secure document storage</li>
    </ul>
</section>
</div>
</main>

<Footer />
</div>
);
};

export default About;

```

src/pages/Contact.jsx

```

import React, { useState } from 'react';
import Header from '../common/Header';
import Footer from '../common/Footer';

const Contact = () => {
    const [formData, setFormData] = useState({
        name: '',
        email: '',
        message: ''
    });
    const [submitted, setSubmitted] = useState(false);

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

    const handleSubmit = (e) => {
        e.preventDefault();
        // In a real app, you would send this data to your backend
        console.log('Contact form submitted:', formData);
        setSubmitted(true);
    };

    return (
        <div className="page">
            <Header />

            <main className="main-content">
                <div className="container">
                    <h1>Contact Us</h1>

                    {submitted ? (
                        <div className="success-message">
                            <p>Thank you for your message! We'll get back to you soon.</p>
                        </div>
                    ) : (
                        <form>
                            <input type="text" name="name" placeholder="Name" value={name} onChange={handleChange} />
                            <input type="email" name="email" placeholder="Email" value={email} onChange={handleChange} />
                            <input type="text" name="message" placeholder="Message" value={message} onChange={handleChange} />
                            <button type="submit" onClick={handleSubmit}>Submit</button>
                        </form>
                    )}
                </div>
            </main>
            <Footer />
        </div>
    );
};

export default Contact;

```

```

        ) : (
            <form onSubmit={handleSubmit} className="contact-form">
                <div className="form-group">
                    <label>Name:</label>
                    <input
                        type="text"
                        name="name"
                        value={formData.name}
                        onChange={handleChange}
                        required
                    />
                </div>

                <div className="form-group">
                    <label>Email:</label>
                    <input
                        type="email"
                        name="email"
                        value={formData.email}
                        onChange={handleChange}
                        required
                    />
                </div>

                <div className="form-group">
                    <label>Message:</label>
                    <textarea
                        name="message"
                        value={formData.message}
                        onChange={handleChange}
                        required
                    />
                </div>

                <button type="submit" className="btn btn-primary">Send Message</button>
            </form>
        )}
    <div className="contact-info">
        <h2>Other Contact Methods</h2>
        <p><strong>Email:</strong> support@moutracker.edu</p>
        <p><strong>Phone:</strong> +1 (123) 456-7890</p>
        <p><strong>Address:</strong> 123 University Ave, Campus City, CC 12345</p>
    </div>
</div>
</main>

<Footer />
</div>
);
};

export default Contact;

```

src/pages/FAQ.jsx

```

import React from 'react';
import Header from '../common/Header';
import Footer from '../common/Footer';

const FAQ = () => {
    const faqs = [
        {
            question: "How do I add a new MOU?",


```

```

        answer: "Navigate to the Dashboard and click on 'Add New MOU'. Fill in all required fields and submit the form."
    },
    {
        question: "Can I download MOU data?",
        answer: "Yes, you can filter and download MOU data in Excel format from the 'Download MOU Data' section."
    },
    {
        question: "How are notifications sent?",
        answer: "Notifications for renewals and monthly activities appear in your Notifications panel and are also sent via email."
    }
];

return (
    <div className="page">
        <Header />

        <main className="main-content">
            <div className="container">
                <h1>Frequently Asked Questions</h1>
                <div className="faq-list">
                    {faqs.map((faq, index) => (
                        <div key={index} className="faq-item">
                            <h3>{faq.question}</h3>
                            <p>{faq.answer}</p>
                        </div>
                    )));
                </div>
            </div>
        </main>

        <Footer />
    </div>
);
};

export default FAQ;

```

src/pages/Home.css

```

/* Hero Section */
.hero-section {
    background: linear-gradient(135deg, #3498db, #2c3e50);
    color: white;
    padding: 5rem 1rem;
    text-align: center;
    position: relative;
    overflow: hidden;
}

.hero-content {
    max-width: 800px;
    margin: 0 auto;
    position: relative;
    z-index: 2;
}

.hero-title {
    font-size: 2.5rem;
    margin-bottom: 1rem;
    font-weight: 700;
    line-height: 1.2;
}

```

```
.hero-subtitle {
  font-size: 1.25rem;
  margin-bottom: 2rem;
  opacity: 0.9;
  max-width: 600px;
  margin-left: auto;
  margin-right: auto;
}

.hero-actions {
  display: flex;
  justify-content: center;
  gap: 1rem;
  margin-top: 2rem;
}

.hero-btn {
  padding: 0.75rem 1.5rem;
  border-radius: 50px;
  font-weight: 600;
  text-decoration: none;
  transition: all 0.3s ease;
  display: inline-block;
}

.hero-btn.primary {
  background-color: white;
  color: #2c3e50;
}

.hero-btn.primary:hover {
  transform: translateY(-3px);
  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
}

.hero-btn.secondary {
  background-color: transparent;
  color: white;
  border: 2px solid white;
}

.hero-btn.secondary:hover {
  background-color: rgba(255, 255, 255, 0.1);
  transform: translateY(-3px);
}

/* Features Section */
.features-section {
  padding: 4rem 1rem;
  background-color: #f8f9fa;
}

.section-container {
  max-width: 1200px;
  margin: 0 auto;
}

.section-title {
  text-align: center;
  font-size: 2rem;
  margin-bottom: 3rem;
  color: #2c3e50;
  position: relative;
}

.section-title::after {
  content: '';
  display: block;
  width: 80px;
```

```
height: 4px;
background: #3498db;
margin: 1rem auto 0;
border-radius: 2px;
}

.features-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
  gap: 2rem;
}

.feature-card {
  background: white;
  padding: 2rem;
  border-radius: 8px;
  text-align: center;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);
}

.feature-card:hover {
  transform: translateY(-10px);
  box-shadow: 0 15px 30px rgba(0, 0, 0, 0.1);
}

.feature-icon {
  font-size: 2.5rem;
  margin-bottom: 1.5rem;
}

.feature-title {
  font-size: 1.25rem;
  margin-bottom: 1rem;
  color: #2c3e50;
}

.feature-description {
  color: #7f8c8d;
  line-height: 1.6;
}

/* Responsive Design */
@media (max-width: 768px) {
  .hero-title {
    font-size: 2rem;
  }

  .hero-subtitle {
    font-size: 1.1rem;
  }

  .hero-actions {
    flex-direction: column;
    align-items: center;
  }

  .hero-btn {
    width: 200px;
    text-align: center;
  }

  .features-grid {
    grid-template-columns: 1fr;
  }
}
```

src/pages/Home.jsx

```
import React from 'react';
import { Link } from 'react-router-dom';
import Header from '../common/Header';
import Footer from '../common/Footer';
import './Home.css';

const Home = ({ isLoggedIn }) => {
  return (
    <div className="home-page">
      <Header isLoggedIn={isLoggedIn} />

      <main className="main-content">
        <section className="hero">
          <div className="container">
            <h1>Automated MOU Data Management</h1>
            <p>Streamline your Memorandum of Understanding tracking process with our comprehensive solution</p>
            {!isLoggedIn && (
              <div className="auth-buttons">
                <Link to="/login" className="btn btn-primary">Login</Link>
                <Link to="/register" className="btn btn-secondary">Register</Link>
              </div>
            )}
          </div>
        </section>

        <section className="features">
          <div className="container">
            <h2>Key Features</h2>
            <div className="feature-grid">
              <div className="feature-card">
                <h3>Easy Data Entry</h3>
                <p>Quickly enter and manage all MOU details in one centralized platform</p>
              </div>
              <div className="feature-card">
                <h3>Smart Filters</h3>
                <p>Download MOU data with powerful filtering options by year, faculty, or institution</p>
              </div>
              <div className="feature-card">
                <h3>Automated Notifications</h3>
                <p>Never miss renewal deadlines with our intelligent reminder system</p>
              </div>
            </div>
          </div>
        </section>
      </main>

      <Footer />
    </div>
  );
};

export default Home;
```

src/pages/Newsroom.jsx

```
import React from 'react';
import Header from '../common/Header';
import Footer from '../common/Footer';

const Newsroom = () => {
  const newsItems = [
    {
      id: 1,
      title: "MOUTracker Launches New Features",
      date: "November 1, 2023",
      summary: "We've added new filtering options and improved the notification system."
    },
    {
      id: 2,
      title: "System Maintenance Scheduled",
      date: "October 15, 2023",
      summary: "The platform will be unavailable on October 20 from 2:00 AM to 4:00 AM for scheduled maintenance."
    }
  ];

  return (
    <div className="page">
      <Header />

      <main className="main-content">
        <div className="container">
          <h1>Newsroom</h1>

          <div className="news-list">
            {newsItems.map(news => (
              <article key={news.id} className="news-item">
                <h2>{news.title}</h2>
                <p className="news-date">{news.date}</p>
                <p className="news-summary">{news.summary}</p>
              </article>
            ))}
          </div>
        </div>
      </main>

      <Footer />
    </div>
  );
};

export default Newsroom;
```

src/services/emailService.js

```
export const sendMonthlyReportEmail = async (recipientEmail) => {
  try {
    // Use the deployed Apps Script URL from environment variables
    const scriptUrl =
      "https://script.google.com/macros/s/AKfycbzRifE_Q5P91CIhJ4j-tRqqgx4SD9j-sJI0cz6sMxdIZfmFKbrlquQvCNrqIKw5yi4Y7w/exec";

    if (!scriptUrl) {
      throw new Error(
        "Apps Script URL not configured in environment variables"
      );
  }
```

```

        }

        const response = await fetch(scriptUrl, {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
            body: JSON.stringify({
                action: "monthly-report",
                email: recipientEmail,
            }),
        });

        if (!response.ok) {
            const errorData = await response.json();
            throw new Error(errorData.message || "Failed to send report");
        }

        return await response.json();
    } catch (error) {
        console.error("Error sending monthly report:", error);
        throw error;
    }
};


```

src/index.jsx

```

import React from 'react';
import ReactDOM from 'react-dom/client';
import './style.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <React.StrictMode>
        <App />
    </React.StrictMode>
);

```

src/App.jsx

```

import React, { useState } from 'react';
import './App.css';
import { BrowserRouter as Router, Routes, Route, Navigate, Link } from 'react-router-dom';
import Home from './components/pages/Home';
import Login from './components/auth/Login';
import Register from './components/auth/Register';
import Dashboard from './components/dashboard/Dashboard';
import About from './components/pages/About';
import FAQ from './components/pages/FAQ';
import Contact from './components/pages/Contact';
import Newsroom from './components/pages/Newsroom';
import MOUForm from './components/dashboard/MOUForm';
import MOUDownload from './components/dashboard/MOUDownload';
import Notifications from './components/dashboard/Notifications';
import MonthlyReport from './components/Dashboard/MonthlyReport';
const App = () => {
    const [isLoggedIn, setIsLoggedIn] = useState(false);

    const handleLoginSuccess = () => {

```

```
        setIsLoggedIn(true);
    };

    const handleLogout = () => {
        setIsLoggedIn(false);
    };

    return (
        <Router>
            <Routes>
                {/* Public Routes */}
                <Route path="/" element={<Home isLoggedIn={isLoggedIn} />} />
                <Route path="/about" element={<About />} />
                <Route path="/faq" element={<FAQ />} />
                <Route path="/contact" element={<Contact />} />
                <Route path="/newsroom" element={<Newsroom />} />
                <Route
                    path="/login"
                    element={
                        isLoggedIn ? (
                            <Navigate to="/dashboard" replace />
                        ) : (
                            <Login onLoginSuccess={handleLoginSuccess} />
                        )
                    }
                />
                <Route
                    path="/register"
                    element={
                        isLoggedIn ? (
                            <Navigate to="/dashboard" replace />
                        ) : (
                            <Register />
                        )
                    }
                />
                {/* Protected Routes */}
                <Route
                    path="/dashboard"
                    element={
                        isLoggedIn ? (
                            <Dashboard onLogout={handleLogout} />
                        ) : (
                            <Navigate to="/login" replace />
                        )
                    }
                />
                <Route
                    path="/mou-form"
                    element={
                        isLoggedIn ? (
                            <MOUForm />
                        ) : (
                            <Navigate to="/login" replace />
                        )
                    }
                />
                <Route
                    path="/mou-download"
                    element={
                        isLoggedIn ? (
                            <MOUDownload />
                        ) : (
                            <Navigate to="/login" replace />
                        )
                    }
                />
                <Route
                    path="/notifications"
```

```

        element={
          isLoggedIn ? (
            <Notifications />
          ) : (
            <Navigate to="/login" replace />
          )
        }
      />

<Route path="/services/monthly-report" element={<MonthlyReport />} />
  </Routes>
</Router>
);
};

export default App;

```

src/main.jsx

```

import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import './index.css'
import App from './App.jsx'

createRoot(document.getElementById('root')).render(
  <StrictMode>
    <App />
  </StrictMode>,
)

```

src/App.css

```

/* General Styles */
body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  line-height: 1.6;
  color: #333;
  background-color: #f5f5f5;
}

.container {
  width: 90%;
  max-width: 1200px;
  margin: 0 auto;
  padding: 0 15px;
}

/* Header Styles */
.header {
  background-color: #2c3e50;
  color: white;
  padding: 1rem 0;
  position: sticky;
  top: 0;
  z-index: 100;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

.header .container {

```

```
        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .logo a {
        color: white;
        text-decoration: none;
        font-size: 1.5rem;
        font-weight: bold;
        display: flex;
        align-items: center;
    }

    .nav-links {
        display: flex;
        list-style: none;
        margin: 0;
        padding: 0;
    }

    .nav-links li {
        margin-left: 1.5rem;
    }

    .nav-links a {
        color: white;
        text-decoration: none;
        font-weight: 500;
        transition: color 0.3s ease;
        padding: 0.5rem 0;
        border-bottom: 2px solid transparent;
    }

    .nav-links a:hover {
        color: #3498db;
        border-bottom-color: #3498db;
    }

/* Button Styles */

    .btn {
        display: inline-block;
        padding: 0.5rem 1rem;
        border-radius: 4px;
        text-decoration: none;
        font-weight: 500;
        text-align: center;
        cursor: pointer;
        transition: all 0.3s ease;
        border: none;
    }

    .btn-primary {
        background-color: #3498db;
        color: white;
    }

    .btn-primary:hover {
        background-color: #2980b9;
        transform: translateY(-2px);
    }

    .btn-secondary {
        background-color: #95a5a6;
        color: white;
    }

    .btn-secondary:hover {
        background-color: #7f8c8d;
    }
```

```
.btn-logout {
  background-color: #e74c3c;
  color: white;
}

.btn-logout:hover {
  background-color: #c0392b;
}

.btn:disabled {
  opacity: 0.6;
  cursor: not-allowed;
}

/* Form Styles */
.auth-container,
.mou-form,
.contact-form {
  max-width: 600px;
  margin: 2rem auto;
  padding: 2rem;
  background: white;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.form-group {
  margin-bottom: 1.5rem;
}

.form-group label {
  display: block;
  margin-bottom: 0.5rem;
  font-weight: 500;
}

.form-group input,
.form-group textarea,
.form-group select {
  width: 100%;
  padding: 0.75rem;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-family: inherit;
  font-size: 1rem;
}

.form-group input:focus,
.form-group textarea:focus,
.form-group select:focus {
  outline: none;
  border-color: #3498db;
  box-shadow: 0 0 0 2px rgba(52,152,219,0.2);
}

.form-group textarea {
  min-height: 120px;
  resize: vertical;
}

.error-message {
  color: #e74c3c;
  margin-bottom: 1rem;
  padding: 0.5rem;
  background-color: #fadbd8;
  border-radius: 4px;
}

.success-message {
```

```
    color: #2ecc71;
    margin-bottom: 1rem;
    padding: 0.5rem;
    background-color: #d5f5e3;
    border-radius: 4px;
}

/* Dashboard Styles */
.dashboard {
    padding: 2rem 0;
}

.dashboard-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 2rem;
    padding-bottom: 1rem;
    border-bottom: 1px solid #eee;
}

.dashboard-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 1.5rem;
}

.dashboard-card {
    padding: 1.5rem;
    background: white;
    border-radius: 8px;
    text-decoration: none;
    color: inherit;
    transition: all 0.3s ease;
    box-shadow: 0 2px 5px rgba(0,0,0,0.05);
    border: 1px solid #eee;
}

.dashboard-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 20px rgba(0,0,0,0.1);
    border-color: #3498db;
}

.dashboard-card h2 {
    color: #2c3e50;
    margin-top: 0;
}

.dashboard-card p {
    color: #7f8c8d;
}

/* Notification Styles */
.notification-list {
    margin-top: 1rem;
}

.notification-item {
    padding: 1rem;
    background: white;
    border-radius: 4px;
    margin-bottom: 0.5rem;
    cursor: pointer;
    transition: all 0.2s ease;
    box-shadow: 0 1px 3px rgba(0,0,0,0.05);
}

.notification-item.unread {
    background-color: #f8f9fa;
```

```
    border-left: 4px solid #3498db;
}

.notification-item.read {
  opacity: 0.8;
  background-color: #f5f5f5;
}

.notification-item:hover {
  background-color: #ebf5fb;
}

.notification-type {
  font-weight: bold;
  text-transform: capitalize;
  color: #2c3e50;
}

.notification-message {
  margin: 0.5rem 0;
}

.notification-date {
  font-size: 0.8rem;
  color: #7f8c8d;
}

/* Home Page Styles */
.hero {
  background-color: #3498db;
  color: white;
  padding: 4rem 0;
  text-align: center;
  background-image: linear-gradient(135deg, #3498db, #2c3e50);
}

.hero h1 {
  font-size: 2.5rem;
  margin-bottom: 1rem;
}

.hero p {
  font-size: 1.2rem;
  margin-bottom: 2rem;
  max-width: 800px;
  margin-left: auto;
  margin-right: auto;
}

.auth-buttons {
  display: flex;
  justify-content: center;
  gap: 1rem;
}

.features {
  padding: 3rem 0;
}

.features h2 {
  text-align: center;
  margin-bottom: 2rem;
  color: #2c3e50;
}

.feature-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  gap: 2rem;
  margin-top: 2rem;
```

```
}

.feature-card {
  padding: 1.5rem;
  background: white;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.05);
  transition: all 0.3s ease;
}

.feature-card:hover {
  transform: translateY(-5px);
  box-shadow: 0 10px 20px rgba(0,0,0,0.1);
}

.feature-card h3 {
  color: #3498db;
  margin-top: 0;
}

/* Footer Styles */
.footer {
  background-color: #34495e;
  color: white;
  padding: 2rem 0;
  margin-top: 2rem;
}

.footer-content {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
  gap: 2rem;
  margin-bottom: 2rem;
}

.footer-section h3 {
  margin-bottom: 1rem;
  color: #ecf0f1;
}

.footer-section ul {
  list-style: none;
  padding: 0;
}

.footer-section ul li {
  margin-bottom: 0.5rem;
}

.footer-section a {
  color: #bdc3c7;
  text-decoration: none;
  transition: color 0.3s ease;
}

.footer-section a:hover {
  color: #3498db;
}

.copyright {
  text-align: center;
  padding-top: 1rem;
  border-top: 1px solid #4a6278;
  color: #bdc3c7;
  font-size: 0.9rem;
}

/* Responsive Styles */
@media (max-width: 768px) {
  .header .container {
```

```

        flex-direction: column;
    }

.nav-links {
    margin-top: 1rem;
}

.nav-links li {
    margin: 0 0.75rem;
}

.feature-grid,
.dashboard-grid {
    grid-template-columns: 1fr;
}

.hero h1 {
    font-size: 2rem;
}
}

/* Utility Classes */
.text-center {
    text-align: center;
}

.mt-1 { margin-top: 0.5rem; }
.mt-2 { margin-top: 1rem; }
.mt-3 { margin-top: 1.5rem; }
.mt-4 { margin-top: 2rem; }

.mb-1 { margin-bottom: 0.5rem; }
.mb-2 { margin-bottom: 1rem; }
.mb-3 { margin-bottom: 1.5rem; }
.mb-4 { margin-bottom: 2rem; }

.loading {
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 2rem;
    color: #7f8c8d;
}

.loading::after {
    content: "";
    display: inline-block;
    width: 1rem;
    height: 1rem;
    border: 2px solid #3498db;
    border-radius: 50%;
    border-top-color: transparent;
    animation: spin 0.8s linear infinite;
    margin-left: 0.5rem;
}

@keyframes spin {
    to { transform: rotate(360deg); }
}

```

src/index.css

```

/* :root {
    font-family: system-ui, Avenir, Helvetica, Arial, sans-serif;
    line-height: 1.5;
}

```

```
font-weight: 400;

color-scheme: light dark;
color: rgba(255, 255, 255, 0.87);
background-color: #242424;

font-synthesis: none;
text-rendering: optimizeLegibility;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}

a {
  font-weight: 500;
  color: #646cff;
  text-decoration: inherit;
}
a:hover {
  color: #535bf2;
}

body {
  margin: 0;
  display: flex;
  place-items: center;
  min-width: 320px;
  min-height: 100vh;
}

h1 {
  font-size: 3.2em;
  line-height: 1.1;
}

button {
  border-radius: 8px;
  border: 1px solid transparent;
  padding: 0.6em 1.2em;
  font-size: 1em;
  font-weight: 500;
  font-family: inherit;
  background-color: #1a1a1a;
  cursor: pointer;
  transition: border-color 0.25s;
}
button:hover {
  border-color: #646cff;
}
button:focus,
button:focus-visible {
  outline: 4px auto -webkit-focus-ring-color;
}

@media (prefers-color-scheme: light) {
  :root {
    color: #213547;
    background-color: #ffffff;
  }
  a:hover {
    color: #747bff;
  }
  button {
    background-color: #f9f9f9;
  }
} */
```

APPSCRIPT CODE

MOU_DETAILS.GS

```
var sheetName = 'Sheet1';
var scriptProp = PropertiesService.getScriptProperties();

function initialSetup() {
  var activeSpreadsheet = SpreadsheetApp.getActiveSpreadsheet();
  scriptProp.setProperty('key', activeSpreadsheet.getId());

  // Add Last Notified column if it doesn't exist
  var sheet = activeSpreadsheet.getSheetByName(sheetName);
  var headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];
  if (headers.indexOf('Last Notified') === -1) {
    sheet.insertColumnAfter(sheet.getLastColumn());
    sheet.getRange(1, sheet.getLastColumn()).setValue('Last Notified');
  }

  // Set up daily trigger for reminders
  createDailyTrigger();
}

function createDailyTrigger() {
  // Remove any existing triggers to avoid duplicates
  var triggers = ScriptApp.getProjectTriggers();
  for (var i = 0; i < triggers.length; i++) {
    if (triggers[i].getHandlerFunction() === 'sendDailyRenewalReminders') {
      ScriptApp.deleteTrigger(triggers[i]);
    }
  }

  // Create new daily trigger (runs every morning at 9 AM)
  ScriptApp.newTrigger('sendDailyRenewalReminders')
    .timeBased()
    .everyDays(1)
    .atHour(9)
    .create();
}

function doPost(e) {
  var lock = LockService.getScriptLock();
  lock.tryLock(10000);

  try {
    // Handle different POST endpoints
    if (e.postData && e.parameter && e.parameter.action === 'monthly-report') {
      return handleMonthlyReportRequest(JSON.parse(e.postData.contents));
    }

    var doc = SpreadsheetApp.openById(scriptProp.getProperty('key'));
    var sheet = doc.getSheetByName(sheetName);
    if (!sheet) throw new Error("Sheet not found: " + sheetName);

    var headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];
    var nextRow = sheet.getLastRow() + 1;

    var fileUrl = '';
    var userEmail = e.parameter['User Email'] || '';

    // Handle base64 file
    if (e.parameter.file) {
      var fileInfo = JSON.parse(e.parameter.file);
      var base64 = fileInfo.base64;
      var fileName = fileInfo.name || "uploaded_file";
      var fileType = fileInfo.type || MimeType.PDF;
    }
  }
}
```

```

var decoded = Utilities.base64Decode(base64);
var blob = Utilities.newBlob(decoded, fileType, fileName);

var folder = DriveApp.getFolderById('1qJyvMmX3VKb7wdH3CpRM5fCnPTqm_co');
var uploadedFile = folder.createFile(blob);
fileUrl = uploadedFile.getUrl();
}

var newRow = headers.map(function (header) {
  var key = header.trim().toLowerCase();
  if (key === 'timestamp') return new Date();
  if (key === 'signed agreement document') return fileUrl;
  return e.parameter[header.trim()] || "";
});

sheet.getRange(nextRow, 1, 1, newRow.length).setValues([newRow]);

// Send confirmation email if user email exists
if (userEmail) {
  var instituteName = e.parameter['Industry/Institute Name'];
  var startDate = e.parameter['Start Date'];
  var endDate = e.parameter['End Date'];

  sendConfirmationEmail(userEmail, instituteName, startDate, endDate);

  // Check if MOU duration is less than 1 month
  var startDateObj = new Date(startDate);
  var endDateObj = new Date(endDate);
  var today = new Date();

  // Calculate duration in milliseconds
  var durationMs = endDateObj - startDateObj;
  var oneMonthMs = 30 * 24 * 60 * 60 * 1000;

  // Check if duration is less than 1 month AND end date is in the future
  if (durationMs < oneMonthMs && endDateObj > today) {
    var subject = 'Immediate MOU Renewal Notice';
    var body = 'Dear User,\n\n' +
      'Your newly created MOU with ' + instituteName + ' has a duration of less than one month.\n\n' +
      'Important Dates:\n' +
      'Start Date: ' + startDate + '\n' +
      'End Date: ' + endDate + '\n\n' +
      'Please note this agreement will expire soon. Consider extending the duration if needed.\n\n' +
      'Thank you,\nMOU Management System';

    MailApp.sendEmail(userEmail, subject, body);
  }
}

return ContentService
  .createTextOutput(JSON.stringify({ result: 'success', row: nextRow }))
  .setMimeType(ContentService.MimeType.JSON);

} catch (error) {
  return ContentService
    .createTextOutput(JSON.stringify({ result: 'error', error: error.message }))
    .setMimeType(ContentService.MimeType.JSON);
} finally {
  lock.releaseLock();
}
}

function handleMonthlyReportRequest(data) {
  try {
    var recipientEmail = data.email;
    var doc = SpreadsheetApp.openById(scriptProp.getProperty('key'));
    var sheet = doc.getSheetByName(sheetName);
  }
}

```

```

// Generate report content
var reportContent = generateMonthlyReportContent(sheet);

// Send email with report
MailApp.sendEmail({
  to: recipientEmail,
  subject: 'Monthly MOU Report - ' + Utilities.formatDate(new Date(),
Session.getScriptTimeZone(), "MMMM yyyy"),
  htmlBody: reportContent.html,
  attachments: [reportContent.pdf]
});

return ContentService.createTextOutput(JSON.stringify({
  success: true,
  message: 'Monthly report sent successfully'
})).setMimeType(ContentService.MimeType.JSON);

} catch (error) {
  return ContentService.createTextOutput(JSON.stringify({
    success: false,
    message: error.message
})).setMimeType(ContentService.MimeType.JSON);
}
}

function generateMonthlyReportContent(sheet) {
  var data = sheet.getDataRange().getValues();
  var headers = data[0];
  var now = new Date();
  var currentMonth = now.getMonth();
  var currentYear = now.getFullYear();

  // Filter for current month's MOUs
  var monthlyData = data.filter(function(row, index) {
    if (index === 0) return false; // Skip header
    var date = new Date(row[headers.indexOf('Start Date')]);
    return date.getMonth() === currentMonth && date.getFullYear() === currentYear;
  });

  // Generate HTML content
  var html = `
    <h1>Monthly MOU Report</h1>
    <p>Generated on: ${Utilities.formatDate(now, Session.getScriptTimeZone(), "MMMM dd, yyyy")}</p>
    <p>Total MOUs this month: ${monthlyData.length}</p>
    <table border="1" style="border-collapse: collapse; width: 100%;">
      <thead>
        <tr style="background-color: #f2f2f2;">
          ${headers.map(header => <th style="padding: 8px; text-align: left;">${header}</th>).join('')}
        </tr>
      </thead>
      <tbody>
        ${monthlyData.map(row => `
          <tr>
            ${row.map(cell => <td style="padding: 8px; border-bottom: 1px solid #ddd;">${cell}</td>).join('')}
          </tr>
        `).join('')}
      </tbody>
    </table>
    <p>This is an automated report. Please contact support if you have any questions.</p>
  `;

  // Generate PDF
  var blob = Utilities.newBlob(html, 'text/html', 'report.html');
  var pdf = blob.getAs('application/pdf').setName('MOU_Report_' + Utilities.formatDate(now, Session.getScriptTimeZone(), "yyyyMM") + '.pdf');

  return {

```

```

        html: html,
        pdf: pdf
    );
}

function sendDailyRenewalReminders() {
    var doc = SpreadsheetApp.openById(scriptProp.getProperty('key'));
    var sheet = doc.getSheetByName(sheetName);
    var data = sheet.getDataRange().getValues();
    var headers = data[0];

    // Find column indexes
    var emailIndex = headers.indexOf('User Email');
    var endDateIndex = headers.indexOf('End Date');
    var instituteIndex = headers.indexOf('Industry/Institute Name');
    var notifiedIndex = headers.indexOf('Last Notified');

    if (emailIndex === -1 || endDateIndex === -1 || instituteIndex === -1) {
        console.error("Required columns not found");
        return;
    }

    var today = new Date();
    today.setHours(0, 0, 0, 0);

    // Process each row
    for (var i = 1; i < data.length; i++) {
        var row = data[i];
        var email = row[emailIndex];
        var endDate = new Date(row[endDateIndex]);
        endDate.setHours(0, 0, 0, 0);
        var institute = row[instituteIndex];

        if (!email || !endDate || isNaN(endDate.getTime())) continue;

        var remainingTime = endDate - today;
        var oneMonthInMs = 30 * 24 * 60 * 60 * 1000;

        // Check if remaining time is less than 1 month and not expired
        if (remainingTime < oneMonthInMs && remainingTime > 0) {
            var remainingDays = Math.ceil(remainingTime / (24 * 60 * 60 * 1000));

            // Determine notification frequency based on urgency
            var shouldNotify = false;
            var subject = '';

            if (remainingDays <= 7) {
                // Daily notifications in last week
                shouldNotify = true;
                subject = 'URGENT: MOU Expiring in ' + remainingDays + ' day' + (remainingDays !== 1 ? 's' : '');
            } else if (remainingDays <= 14) {
                // Every 3 days in second week
                var lastNotified = notifiedIndex !== -1 ? new Date(row[notifiedIndex]) : null;
                if (!lastNotified || (today - lastNotified) >= 3 * 24 * 60 * 60 * 1000) {
                    shouldNotify = true;
                    subject = 'MOU Expiring in ' + remainingDays + ' days';
                }
            } else {
                // Weekly until 2 weeks before
                var lastNotified = notifiedIndex !== -1 ? new Date(row[notifiedIndex]) : null;
                if (!lastNotified || (today - lastNotified) >= 7 * 24 * 60 * 60 * 1000) {
                    shouldNotify = true;
                    subject = 'MOU Expiring in ' + remainingDays + ' days';
                }
            }

            if (shouldNotify) {
                var body = 'Dear User,\n\n' +

```

```

        'Your MOU with ' + institute + ' has ' + remainingDays + ' day' + (remainingDays !== 1
? 's' : '') + ' remaining.\n\n' +
        'Expiration Date: ' + endDate.toDateString() + '\n\n' +
        (remainingDays <= 7 ? 'URGENT: ' : '') + 'Please take necessary action to renew this
agreement.\n\n' +
        'Thank you,\nMOU Management System';

    MailApp.sendEmail(email, subject, body);

    // Update last notified date if column exists
    if (notifiedIndex !== -1) {
        sheet.getRange(i + 1, notifiedIndex + 1).setValue(today);
    }
}
}

function sendConfirmationEmail(email, instituteName, startDate, endDate) {
    var subject = 'MOU Submission Confirmation';
    var body = 'Dear User,\n\n' +
        'Your MOU with ' + instituteName + ' has been successfully submitted.\n\n' +
        'Details:\n' +
        'Start Date: ' + startDate + '\n' +
        'End Date: ' + endDate + '\n\n' +
        'Thank you for using our MOU management system.';

    MailApp.sendEmail(email, subject, body);
}

function sendMonthlyReminders() {
    var doc = SpreadsheetApp.openById(scriptProp.getProperty('key'));
    var sheet = doc.getSheetByName(sheetName);
    var data = sheet.getDataRange().getValues();
    var headers = data[0];

    // Find column indexes
    var emailIndex = headers.indexOf('User Email');
    var endDateIndex = headers.indexOf('End Date');
    var instituteIndex = headers.indexOf('Industry/Institute Name');

    if (emailIndex === -1 || endDateIndex === -1 || instituteIndex === -1) return;

    var today = new Date();
    var oneMonthLater = new Date();
    oneMonthLater.setMonth(oneMonthLater.getMonth() + 1);

    // Process each row
    for (var i = 1; i < data.length; i++) {
        var row = data[i];
        var email = row[emailIndex];
        var endDate = new Date(row[endDateIndex]);
        var institute = row[instituteIndex];

        if (!email || !endDate || isNaN(endDate.getTime())) continue;

        // Check for renewals (expiring in 1 month)
        if (endDate <= oneMonthLater && endDate >= today) {
            var subject = 'MOU Renewal Reminder';
            var body = 'Dear User,\n\n' +
                'Your MOU with ' + institute + ' is expiring soon on ' + endDate.toDateString() +
                '.\n\n' +
                'Please consider renewing the agreement if needed.\n\n' +
                'Thank you,\nMOU Management System';

            MailApp.sendEmail(email, subject, body);
        }
    }

    // Send monthly activity reminder (to all users)
    var activitySubject = 'Monthly MOU Activity Reminder';
}

```

```

    var activityBody = 'Dear User,\n\n' +
      'This is a reminder to submit your monthly activities related to your MOUs.\n\n' +
      'Please log in to the system and update your activities for this month.\n\n' +
      'Thank you,\nMOU Management System';

    MailApp.sendEmail(email, activitySubject, activityBody);
  }
}

function doGet(e) {
  const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet1');
  const data = sheet.getDataRange().getValues();
  const headers = data.shift();

  const jsonData = data.map(row => {
    const obj = {};
    headers.forEach((key, i) => {
      obj[key] = row[i];
    });
    return obj;
  });

  return ContentService
    .createTextOutput(JSON.stringify({ data: jsonData }))
    .setMimeType(ContentService.MimeType.JSON);
}

```

MOU_LOGIN.GS

```

// Updated with proper CORS handling and data validation
var scriptProp = PropertiesService.getScriptProperties();
var sheetName = 'Sheet1';

function doGet(e) {
  return handleRequest(e);
}

function doPost(e) {
  return handleRequest(e);
}

function doOptions() {
  return buildCorsResponse();
}

function handleRequest(e) {
  console.log('Incoming request:', JSON.stringify(e));
  try {
    const data = e.postData ? JSON.parse(e.postData.contents) : {};
    const ss = SpreadsheetApp.openById(scriptProp.getProperty('spreadsheetId'));
    const sheet = ss.getSheetByName(sheetName);
    const rows = sheet.getDataRange().getValues();

    let response;
    switch(data.action) {
      case 'register':
        response = handleRegistration(data, rows, sheet);
        break;
      case 'login':
        response = handleLogin(data, rows);
        break;
      default:
        response = { success: false, message: 'Invalid action' };
    }
  }
}

```

```

        return buildCorsResponse(response);
    } catch (error) {
        return buildCorsResponse({
            success: false,
            message: 'Server error: ' + error.message
        }, 500);
    }
}

function handleRegistration(data, rows, sheet) {
    const cleanEmail = data.email.toLowerCase().trim();
    const cleanName = data.name.trim();
    const cleanPassword = data.password.trim();

    // Debug: Log what we're checking
    console.log(`Checking registration for: ${cleanEmail}`);
    console.log('First 3 rows:', rows.slice(0, 3));

    const existingUser = rows.slice(1).find(row => {
        const storedEmail = row[1] ? .toString().toLowerCase().trim();
        return storedEmail === cleanEmail;
    });

    if (existingUser) {
        return {
            success: false,
            message: 'Email already registered',
            debug: {
                input: { email: cleanEmail, name: cleanName },
                existing: { email: existingUser[1], name: existingUser[0] }
            }
        };
    }

    sheet.appendRow([cleanName, cleanEmail, cleanPassword]);

    return {
        success: true,
        message: 'Registration successful',
        debug: {
            input: { email: cleanEmail, name: cleanName },
            registered: true
        }
    };
}

function handleLogin(data, rows) {
    const cleanEmail = data.email.toLowerCase().trim();
    const cleanPassword = data.password.trim();

    // Debug: Log what we're comparing
    console.log(`Comparing: ${cleanEmail} | ${cleanPassword}`);
    console.log('First 3 rows:', rows.slice(0, 3));

    const user = rows.slice(1).find(row => {
        const storedEmail = row[1] ? .toString().toLowerCase().trim();
        const storedPassword = row[2] ? .toString().trim();

        console.log(`Stored: ${storedEmail} | ${storedPassword}`);
        return storedEmail === cleanEmail && storedPassword === cleanPassword;
    });

    return {
        success: !!user,
        message: user ? 'Login successful' : 'Invalid credentials',
        debug: {
            input: { email: cleanEmail, password: cleanPassword },
            found: user ? { email: user[1], password: user[2] } : null
        }
    };
}

```

```

}

// function buildCorsResponse(data = {}, statusCode = 200) {
//   return HtmlService.createHtmlOutput(JSON.stringify(data))
//     .setSandboxMode(HtmlService.SandboxMode.IFRAME)
//     .setXFrameOptionsMode(HtmlService.XFrameOptionsMode.ALLOWALL)
//     .setContent(JSON.stringify(data))
//     .setStatusCode(statusCode);
// }
function buildCorsResponse(data = {}, statusCode = 200) {
  const output = ContentService.createTextOutput(JSON.stringify(data));

  return output;
}

```

Index.html

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>

```

Server.JS

```

import express from "express";
import cors from "cors";
import fetch from "node-fetch";

const app = express();
const PORT = process.env.PORT || 3000;

// Enhanced CORS configuration
app.use(
  cors({
    origin: true,
    methods: ["GET", "POST", "OPTIONS"],
    allowedHeaders: ["Content-Type", "Authorization"],
    credentials: true,
  })
);

app.use(express.json());

// Handle OPTIONS requests first
app.options("/proxy", cors());

app.all("/proxy", async (req, res) => {
  try {
    if (!req.query.url) {
      return res.status(400).json({ error: "Missing target URL" });
    }
  }
});

```

```

const targetUrl = decodeURIComponent(req.query.url);

const fetchOptions = {
  method: req.method,
  headers: {
    "Content-Type": "application/json",
    Accept: "application/json",
  },
  redirect: "follow",
};

// Only include body for POST requests
if (req.method === "POST") {
  fetchOptions.body = JSON.stringify(req.body);
}

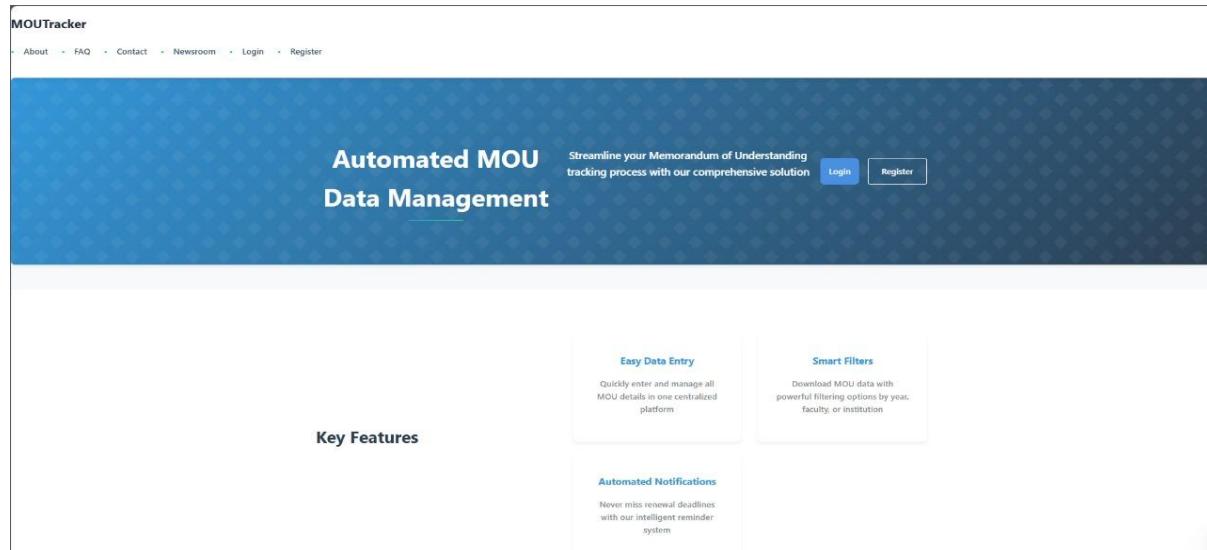
const response = await fetch(targetUrl, fetchOptions);

// Handle non-JSON responses
const contentType = response.headers.get("content-type");
if (contentType && contentType.includes("application/json")) {
  const data = await response.json();
  return res.status(response.status).json(data);
} else {
  const text = await response.text();
  return res.status(response.status).send(text);
}
} catch (error) {
  console.error("Proxy error:", error);
  return res.status(500).json({
    success: false,
    message: "Proxy server error: " + error.message,
  });
}
});

app.listen(PORT, () => {
  console.log(`Proxy server running on port ${PORT}`);
});

```

RESULT WITH SCREENSHOTS:



MOUTracker

~ About ~ FAQ ~ Contact ~ Newsroom ~ Login ~ Register

About MOUTracker

Our Mission

MOUTracker was developed to streamline the management of Memorandums of Understanding between educational institutions and industry partners. Our goal is to automate the tracking process and ensure no important deadlines are missed.

Features

- Centralized MOU data management
- Automated renewal reminders
- Easy data entry and retrieval
- Customizable filtering options
- Secure document storage

Quick Links

- Home
- About
- FAQ

Contact

Email:
contact@moutracker.edu

Phone: +1 (123) 456-7890

© 2023 MOUTracker. All rights reserved.

MOUTracker

~ About ~ FAQ ~ Contact ~ Newsroom ~ Login ~ Register

Frequently Asked Questions

Q How do I add a new MOU?

A Navigate to the Dashboard and click on 'Add New MOU'. Fill in all required fields and submit the form.

Q Can I download MOU data?

A Yes, you can filter and download MOU data in Excel format from the 'Download MOU Data' section.

Q How are notifications sent?

A Notifications for renewals and monthly activities appear in your Notifications panel and are also sent via email.

Quick Links

- Home
- About
- FAQ

Contact

Email:
contact@moutracker.edu

Phone: +1 (123) 456-7890

© 2023 MOUTracker. All rights reserved.

MOUTracker

About FAQ Contact Newsroom Login Register

Contact Us

Name: Email:

Message:

Send Message

Other Contact Methods

Email: support@moutracker.edu
Phone: +1 (123) 456-7890
Address: 123 University Ave, Campus City, CC 12345

Quick Links

- Home
- About
- FAQ

Contact

Email: contact@moutracker.edu
Phone: +1 (123) 456-7890

© 2023 MOUTracker. All rights reserved.

MOUTracker

About FAQ Contact Newsroom Login Register

Contact Us

Thank you for your message! We'll get back to you soon.

Other Contact Methods

Email: support@moutracker.edu
Phone: +1 (123) 456-7890
Address: 123 University Ave, Campus City, CC 12345

Quick Links

- Home
- About
- FAQ

Contact

Email: contact@moutracker.edu
Phone: +1 (123) 456-7890

© 2023 MOUTracker. All rights reserved.

MOUTracker

About FAQ Contact Newsroom Login Register

Newsroom

MOUTracker Launches New Features
November 1, 2023
We've added new filtering options and improved the notification system.

System Maintenance Scheduled
October 15, 2023
The platform will be unavailable on October 20 from 2:00 AM to 4:00 AM for scheduled maintenance.

Quick Links

- Home
- About
- FAQ

Contact

Email: contact@moutracker.edu
Phone: +1 (123) 456-7890

© 2023 MOUTracker. All rights reserved.

MOUTracker

About + FAQ + Contact + Newsroom + Login + Register

Login

Email
disha2210204@ssr

Password
.....

[Login](#)

New user? [Register](#)

MOUTracker

About + FAQ + Contact + Newsroom + Login + Register

Register

Name
Hannah

Email
hannah2210483@ssr

Password
.....

Confirm Password
.....

[Register](#)

Already have an account? [Login](#)

MOUTracker

About • FAQ • Contact • Newsroom • Login • Register

Register

Registration successful

Name Email

Password Confirm Password

[Register](#)

Already have an account? [Login](#)

MOUTracker

About • FAQ • Contact • Newsroom • Login • Register

Dashboard

[Logout](#)

[Add New MOU](#)

Enter details of a new Memorandum of Understanding

[Download MOU Data](#)

Filter and download existing MOU information

[Notifications](#)

View alerts and reminders

Quick Links

- Home
- About
- FAQ

Contact

Email:
contact@moutracker.edu

Phone: +1 (123) 456-7890

© 2025 MOUTracker. All rights reserved.

MOUTracker

About FAQ Contact Newsroom Login Register

MOU Submission Form

Industry/Institute Name

Start Date of MOU

End Date of MOU

Upload Signed Agreement Document MOU.pdf

Faculty Name

Faculty Details

Academic Year

Purpose of MOU

Outcomes of MOU

Quick Links

- Home
- About
- FAQ

Contact

Email: contact@moutracker.edu
Phone: +1 (023) 456-7890

© 2023 MOUTracker. All rights reserved.

MOUTracker

About FAQ Contact Newsroom Login Register

MOU Submission Form

✓ MOU submitted successfully!

Industry/Institute Name

Start Date of MOU

End Date of MOU

Upload Signed Agreement Document MOU.pdf

Faculty Name

Faculty Details

Academic Year

Purpose of MOU

Outcomes of MOU

Quick Links

- Home
- About
- FAQ

Contact

Email: contact@moutracker.edu
Phone: +1 (023) 456-7890

© 2023 MOUTracker. All rights reserved.

MOUTracker

- About
- FAQ
- Contact
- Newsroom
- Login
- Register

Filter & Download MOU Data

Academic Year:	Industry/Institute:	Faculty Name:	Duration:
2025-2026	SSN	<input type="text" value="Search..."/>	All

[Download CSV](#)

1 entries found.

Quick Links

- Home
- About
- FAQ

Contact

Email: contact@moutracker.edu
Phone: +1 (123) 456-7890

© 2025 MOUTracker. All rights reserved.

A	B	C	D	E	F	G	H	I	J	K
Industry/Institute	Start Date	End Date	Signed Ago	Faculty Name	Faculty Designation	Academic Year	Purpose of MOU	User Email		
SSN	2024-05-15	2024-06-01	https://dri.ac.in	Zara	Sundar	2025-2026	Project	dilsha2210204@ssn.edu.in		

MOUTracker

- About
- FAQ
- Contact
- Newsroom
- Login
- Register

Email Notification Preferences

Manage how you receive notifications about MOUs

New MOU Entries Receive emails when new MOUs are added

Renewal Reminders Get reminders before MOUs expire

Monthly Summary Receive a monthly report of all MOUs

Notifications will be sent to: hannah2210480@sn.edu.in

[Save Preferences](#)

Quick Links

- Home
- About
- FAQ

Contact

Email: contact@moutracker.edu
Phone: +1 (123) 456-7890

© 2025 MOUTracker. All rights reserved.

filtered_mou_data

	A	B	C	D	E	F	G	H	I	J
1	Industry/Institute	Start Date	End Date	Signed Agreement	Faculty Name	Faculty Details	Academic Year	Purpose of MOU	User Email	Last Notified
2	SSN	2025-05-10	2026-05-10	https://drive.google.com	Sundar	2022-2023	Project	dilsha2210204@ssn.edu.in		
3	SSN	2025-05-10	2026-05-10	https://drive.google.com	Rohan	2023-2024	Collaboration	dilsha2210204@ssn.edu.in		
4	SSN	2025-05-10	2027-05-10	https://drive.google.com	Lavanya	2023-2024	research	dilsha2210204@ssn.edu.in		
5	SSN	2024-06-12	2027-06-12	https://drive.google.com	asdf	2022-2023	adf			dillysheila@gmail.com

MOU-Login

	A	B	C	D	E
1	name	email	password		
2	Dilsha Singh D	dilsha2210204@ssn.edu.in	789456123		
3	Dillu	dilshasingh@gmail.com	123456789		
4	Dilsha	dilsha2210204@ssn.edu.in	789456123		
5	rohan	r@gamil.com	789456123		



MOU Submission Confirmation

Inbox



dilsha2210204@ssn.... 10:08 AM
to me ▾

Dear User,

Your MOU with Tazzle has been successfully submitted.

Details:

Start Date: 2025-04-15

End Date: 2025-05-10

Thank you for using our MOU management system.

--
::DISCLAIMER::

The contents of this e-mail and any attachment(s) are confidential and intended for the named recipient(s) only. Views or opinions, if any, presented in this email are solely those of the author and may not necessarily reflect the views or opinions of SSN Institutions (SSN) or its affiliates. Any form of reproduction, dissemination, copying, disclosure, modification, distribution and / or publication of this message without the prior written consent of authorized representative of SSN is strictly prohibited. If you have received this email in error please delete it and



Immediate MOU Renewal Notice

Inbox



dilsha2210204@ssn.... 10:08 AM
to me ▾

Dear User,

Your newly created MOU with Tazzle has a duration of less than one month.

Important Dates:

Start Date: 2025-04-15

End Date: 2025-05-10

Please note this agreement will expire soon. Consider extending the duration if needed.

Thank you,
MOU Management System

--
::DISCLAIMER::

The contents of this e-mail and any attachment(s) are confidential and intended for the named recipient(s) only. Views or opinions, if any, presented in this email are solely those of the author and may not necessarily reflect the views or opinions of SSN Institutions (SSN) or its affiliates. Any form of reproduction, dissemination, copying, disclosure, modification, distribution and / or publication of this message without the prior written consent of authorized representative of SSN is strictly prohibited. If you have received this email in error please delete it and



LEARNING OUTCOMES:

1. We learnt to use Node.js and Google Apps Script to implement secure user registration and authentication.
 2. Gained experience in building a responsive React frontend connected to backend scripts for data operations.
 3. Understood how to automate email notifications for MOU reminders, updates, and monthly activities.
 4. Explored techniques to dynamically update and manage structured data using Excel sheets via Apps Script.
 5. Strengthened our full-stack development skills by creating an end-to-end automated MOU tracking system.
-