```
import React, { useState } from 'react';
import axios from 'axios';
import './EmailTemplateModal.css';

const EmailTemplateModal = ({ onClose }) => {
  const [emailTemplate, setEmailTemplate] = useState({
    mailAlertName: '',
    description: '',
    from: '',
    to: '',
    cc: '',
    bcc: '',
    subject: '',
    message: ''
  });
  const [attachment, setAttachment] = useState(null);

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

  const handleFileChange = (event) => {
    setAttachment(event.target.files[0]);
  };



const handleSubmit = async (event) => {
    event.preventDefault();
```

```
const formData = new FormData();
formData.append('from', emailTemplate.from);
formData.append('to', emailTemplate.to);
formData.append('cc', emailTemplate.cc);
formData.append('bcc', emailTemplate.bcc);
formData.append('subject', emailTemplate.subject);
formData.append('message', emailTemplate.message);
if (attachment) {
  formData.append('attachment', attachment);
}

try {
  const response = await axios.post('http://localhost:8086/mail/send', formData, {
    headers: {
      'Content-Type': 'multipart/form-data'
    }
  });
  console.log('Email sent:', response.data);
  alert('Email sent successfully!');
  onClose();
} catch (error) {
  console.error('Error sending email:', error);
  if (error.response) {
    console.error('Error data:', error.response.data);
    console.error('Error status:', error.response.status);
    console.error('Error headers:', error.response.headers);
    alert(`Error sending email: ${error.response.data.message || error.response.statusText}`);
  } else if (error.request) {
    console.error('Error request:', error.request);
    alert('No response received from the server. Please check if the server is running.');
  } else {
   console.error('Error message:', error.message);
   alert(`Error: ${error.message}`);
  }
 }
};
return (
 <div className="modal-overlay">
  <div className="modal-content">
   <h2>Edit-Timesheet approval mail template</h2>
   <form onSubmit={handleSubmit}>
    <div>
     <label htmlFor="mailAlertName">Mail Alert Name</label>
     <input
      type="text"
      id="mailAlertName"
      name="mailAlertName"
      value={emailTemplate.mailAlertName}
      onChange={handleChange}
     />
    </div>
    <div>
     <label htmlFor="description">Description</label>
     <textarea
      id="description"
      name="description"
      value={emailTemplate.description}
      onChange={handleChange}
     ></textarea>
    </div>
    <div>
     <label htmlFor="from">From</label>
     <input
      type="text"
      id="from"
      name="from"
      value={emailTemplate.from}
      onChange={handleChange}
     />
    </div>
    <div>
     <label htmlFor="to">To</label>
     <input
      type="text"
      id="to"
      name="to"
      value={emailTemplate.to}
      onChange={handleChange}
     />
    </div>
    <div>
     <label htmlFor="cc">CC</label>
     <input
      type="text"
      id="cc"
      name="cc"
      value={emailTemplate.cc}
      onChange={handleChange}
     />
    </div>
    <div>
     <label htmlFor="bcc">BCC</label>
```

```
  <input
    type="text"
    id="bcc"
    name="bcc"
    value={emailTemplate.bcc}
    onChange={handleChange}
  />
</div>
<div>
  <label htmlFor="subject">Subject</label>
  <input
    type="text"
    id="subject"
    name="subject"
    value={emailTemplate.subject}
    onChange={handleChange}
  />
</div>
<div>
  <label htmlFor="message">Message</label>
  <textarea
    id="message"
    name="message"
    value={emailTemplate.message}
    onChange={handleChange}
  ></textarea>
</div>
<div>
  <label htmlFor="attachment">Attachment</label>
  <input
    type="file"
```

```
              id="attachment"
              name="attachment"
              onChange={handleFileChange}
            />
          </div>
          <div className="button-group">
            <button type="submit">Save</button>
            <button type="button" onClick={onClose}>Cancel</button>
          </div>
        </form>
      </div>
    </div>
  );
};

export default EmailTemplateModal;
package com.Texsmartly.Time.Sheet.Model;

import jakarta.validation.constraints.Email;
import lombok.*;
import org.springframework.data.mongodb.core.mapping.Document;

@Setter
@Getter
@Document(collection = "mail-structure")
@Data
@AllArgsConstructor
@NoArgsConstructor

public class MailStructure {
    private String from;
    @Email(message = "Email should be a valid email")
    private String to;
    private String cc;
    private String bcc;
    private String subject;
    private String message;
    private byte[] attachment;

}
```