

ContactMailer CheetSheet

1. ***Rails new app-name --api --database=postgresql***

2. **config/database.yml**

```
`` default: &default
adapter: postgresql
encoding: unicode
username: your_username
password: your_password
pool: 5
timeout: 5000

development:
<<: *default
database: your_app_name_development

test:
<<: *default
database: your_app_name_test

production:
<<: *default
database: your_app_name_production
username: <%= ENV['DATABASE_USERNAME'] %>
password: <%= ENV['DATABASE_PASSWORD'] %>
``
```

3. ***rails active_storage:install***

4. ***rails generate mailer ContactMailer***

contact_mailer.rb

```
class ContactMailer < ApplicationMailer
  def contact_email(name, email, message, attachment)
    @name = name
    @email = email
    @message = message

    if attachment.present?
      attachments[attachment.original_filename] = {
        mime_type: attachment.content_type,
        content: attachment.read
      }
    end
  end
end
```

```

    mail(to: 'example@gmail.com', subject: 'Contact Form Inquiry')
  end
end

```

5. rails generate controller ContactForm

contact_form_controller.rb

```

class ContactFormController < ApplicationController
  def create
    name = params[:name]
    email = params[:email]
    message = params[:message]
    attachment = params[:attachment]

    ContactMailer.contact_email(name, email, message, attachment).deliver_now

    render json: { status: 'success' }
  end
end

```

6. config/routes.rb

```

Rails.application.routes.draw do
  get "up" => "rails/health#show", as: :rails_health_check

  resources :contact_form, only: [:create], controller: 'contact_form'
end

```

7. rails db:create rails db:migrate

8. Frontend – Creating ContactForm

```

import React, { useState } from 'react';
import Swal from 'sweetalert2';

function Contact() {
  const [formData, setFormData] = useState({
    name: "",
    email: "",
    message: "",
    attachment: null,
  });

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

```

```

    ...formData,
    [name]: value,
  });
};

const handleFileChange = (e) => {
  setFormData({
    ...formData,
    attachment: e.target.files[0],
  });
};

const handleSubmit = async (e) => {
  e.preventDefault();
  const formDataObj = new FormData();
  formDataObj.append('name', formData.name);
  formDataObj.append('email', formData.email);
  formDataObj.append('message', formData.message);
  formDataObj.append('attachment', formData.attachment);

  try {
    const response = await fetch('http://localhost:3000/contact_form', {
      method: 'POST',
      body: formDataObj,
    });

    if (!response.ok) {
      throw new Error('Network response was not ok ' + response.statusText);
    }

    const result = await response.json();
    console.log("Email sent successfully:", result);
    Swal.fire({
      icon: "success",
      title: "Success",
    });
    e.target.reset();
  } catch (error) {
    console.error("Error sending email:", error);
    Swal.fire({
      icon: "error",
      title: "Wrong",
      text: error.message,
    });
  }
};

```

```

return (
  <form onSubmit={handleSubmit}>
    <input type="text" name="name" value={formData.name} onChange={handleChange}
    placeholder="Name" required />
    <input type="email" name="email" value={formData.email} onChange={handleChange}
    placeholder="Email" required />
    <textarea name="message" value={formData.message} onChange={handleChange}
    placeholder="Message" required></textarea>
    <input type="file" onChange={handleFileChange} required /> <button
    type="submit">Submit</button>
  </form>
); }

```

export default Contact;

9. Frontend - package.json:

"proxy": <http://localhost:3000>,

10. Config/environments/production.rb

config.assets.compile = false

11. Gemfile

```

group :development do
  gem 'dotenv-rails'
end

```

gem 'rack-cors', require: 'rack/cors'

Bundle install

12. Create .env file in main folder write your passwords there.

DATABASE_PASSWORD='xxx'

Add to .gitignore:

.env

13. config/application.rb

```

config.middleware.insert_before 0, Rack::Cors do
  allow do
    origins 'http://localhost:3001'
    resource '*',
      headers: :any,

```

```
    methods: [:get, :post, :put, :patch, :delete, :options, :head],
    credentials: true
  end end
```

14. Need smtp service example mailtrap

```
config.action_mailer.delivery_method = :smtp
config.action_mailer.smtp_settings = {
  :user_name => ENV['MAILTRAP_USERNAME'],
  :password => ENV['MAILTRAP_PASSWORD'],
  :address => 'sandbox.smtp.mailtrap.io',
  :host => 'sandbox.smtp.mailtrap.io',
  :port => '2525',
  :authentication => :cram_md5
}
```

15. app/views/contact_mailer/contact_email.html.erb

Template for the email you will receive.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Contact Form Submission</title>
</head>
<body>
  <p>Name: <%= @name %></p>
  <p>Email: <%= @email %></p>
  <p>Message:</p>
  <p><%= @message %></p>
</body>
</html>
```