

草月いけばな展 花材注文フォーム - 全ソースコード

Sogetsu Exhibition Flower Order Form - Complete Source Code

このファイルには、プロジェクトの全ソースコードが含まれています。

コピー&ペーストで使用できます。

プロジェクト構造 / Project Structure

...

flowerorderform/

├── package.json

├── vite.config.ts

├── tsconfig.json

├── tsconfig.app.json

├── tsconfig.node.json

├── tailwind.config.js

├── postcss.config.js

├── eslint.config.js

├── index.html

├── README.md

├── SETUP.md

├── CONTRIBUTING.md

├── LICENSE

└── CHANGELOG.md

```
└── src/
    ├── main.tsx
    ├── App.tsx
    ├── index.css
    ├── vite-env.d.ts
    ├── translations.ts
    └── components/
        ├── FormField.tsx
        ├── ImageUpload.tsx
        └── LanguageToggle.tsx
```

...

package.json

```
```json
```

```
{
 "name": "sogetsu-flower-order-form",
 "private": true,
 "version": "1.0.0",
 "description": "Bilingual flower material order form for the 106th Sogetsu Annual Exhibition",
 "type": "module",
 "homepage": "https://bloemism.github.io/flowerorderform",
 "repository": {
 "type": "git",
```

```
"url": "https://github.com/bloemism/flowerorderform.git"

},

"keywords": [

 "sogetsu",

 "ikebana",

 "flower",

 "order-form",

 "react",

 "typescript",

 "bilingual",

 "mobile-responsive"

],

"author": "Bloomwinkel",

"license": "MIT",

"scripts": {

 "dev": "vite",

 "build": "vite build",

 "lint": "eslint .",

 "preview": "vite preview"

},

"dependencies": {

 "lucide-react": "^0.344.0",

 "react": "^18.3.1",

 "react-dom": "^18.3.1"

},

"devDependencies": {

 "@eslint/js": "^9.9.1",
```

```
"@types/react": "^18.3.5",
"@types/react-dom": "^18.3.0",
"@vitejs/plugin-react": "^4.3.1",
"autoprefixer": "^10.4.18",
"eslint": "^9.9.1",
"eslint-plugin-react-hooks": "^5.1.0-rc.0",
"eslint-plugin-react-refresh": "^0.4.11",
"globals": "^15.9.0",
"postcss": "^8.4.35",
"tailwindcss": "^3.4.1",
"typescript": "^5.5.3",
"typescript-eslint": "^8.3.0",
"vite": "^5.4.2"
}
}
```
---
```

index.html

```
```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>Sogetsu Exhibition Flower Order Form</title>

</head>

<body>

 <div id="root"></div>

 <script type="module" src="/src/main.tsx"></script>

</body>

</html>

...

```

```
src/main.tsx
```

```
```tsx

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

createRoot(document.getElementById('root')!).render(

  <StrictMode>

    <App />

  </StrictMode>

);

...

---
```

src/index.css

```css

@tailwind base;

@tailwind components;

@tailwind utilities;

```

src/App.tsx

```tsx

import React, { useState } from 'react';

import { Flower2, Mail, Calendar, Clock, User, MapPin, Palette, DollarSign, FileText } from 'lucide-react';

import LanguageToggle from './components/LanguageToggle';

import ImageUpload from './components/ImageUpload';

import FormField from './components/FormField';

import { translations } from './translations';

function App() {

const [language, setLanguage] = useState<'en' | 'ja'>('en');

const [formData, setFormData] = useState({

name: "",

```
 phone: "",
 email: "",
 exhibitorNumber: "",
 period: "",
 seatNumber: "",
 flowerType: "",
 quantity: "",
 colors: "",
 budget: "",
 deliveryDate: "",
 deliveryTime: "",
 specialRequests: ""
 });

const [images, setImages] = useState<File[]>([]);

const t = translations[language];

const handleInputChange = (field: string, value: string) => {
 setFormData(prev => ({ ...prev, [field]: value }));
};

const handleSubmit = (e: React.FormEvent) => {
 e.preventDefault();

 // Here you would typically send the form data and images to your backend

 console.log('Form data:', formData);
 console.log('Images:', images);

 alert(language === 'en' ? 'Order submitted successfully!' : '注文が正常に送信されました！');
```

```
};
```

```
return (
```

```
<div className="min-h-screen bg-gradient-to-br from-green-50 to-emerald-100">
```

```
 { /* Header */ }
```

```
<div className="bg-white shadow-sm border-b border-green-100">
```

```
<div className="max-w-4xl mx-auto px-4 py-4 flex justify-between items-center">
```

```
<div className="flex items-center gap-3">
```

```
<Flower2 className="w-8 h-8 text-green-600" />
```

```
<div>
```

```
<h1 className="text-lg font-bold text-gray-800 leading-tight">
```

```
 {t.title}
```

```
</h1>
```

```
<p className="text-sm text-green-600 font-medium">{t.subtitle}</p>
```

```
</div>
```

```
</div>
```

```
<LanguageToggle language={language} onToggle={setLanguage} />
```

```
</div>
```

```
</div>
```

```
 { /* Main Content */ }
```

```
<div className="max-w-4xl mx-auto px-4 py-8">
```

```
 { /* Contact Information */ }
```

```
<div className="bg-white rounded-xl shadow-sm p-6 mb-8 border border-green-100">
```

```
<div className="flex items-start gap-4">
```

```
<Mail className="w-6 h-6 text-green-600 mt-1 flex-shrink-0" />
```

```
<div className="space-y-2">
```



```

 <h2 className="text-lg font-semibold text-gray-800">{t.contact}</h2>

 <p className="text-green-600 font-medium">{t.email}</p>

 <div className="bg-blue-50 p-4 rounded-lg border-l-4 border-blue-400">

 <p className="text-sm text-blue-800 mb-2">{t.deadline}</p>

 <p className="text-sm text-blue-700">{t.payment}</p>

 </div>

 </div>

</div>

</div>

</div>

{ /* Order Form */ }

<form onSubmit={handleSubmit} className="space-y-8">

 { /* Personal Information */ }

 <div className="bg-white rounded-xl shadow-sm p-6 border border-green-100">

 <div className="flex items-center gap-3 mb-6">

 <User className="w-6 h-6 text-green-600" />

 <h3 className="text-lg font-semibold text-gray-800">{t.personalInfo}</h3>

 </div>

 <div className="grid grid-cols-1 md:grid-cols-2 gap-6">

 <FormField label={t.name} required>

 <input

 type="text"

 value={formData.name}

 onChange={(e) => handleInputChange('name', e.target.value)}

 placeholder={t.namePlaceholder}

```

```
 className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-green-500 focus:border-green-500 transition-all"

 required

 />

</FormField>
```

```
<FormField label={t.phone} required>

 <input

 type="tel"

 value={formData.phone}

 onChange={(e) => handleInputChange('phone', e.target.value)}

 placeholder={t.phonePlaceholder}

 className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-green-500 focus:border-green-500 transition-all"

 required

 />

</FormField>
```

```
<FormField label={t.emailField} required>

 <input

 type="email"

 value={formData.email}

 onChange={(e) => handleInputChange('email', e.target.value)}

 placeholder={t.emailPlaceholder}

 className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-green-500 focus:border-green-500 transition-all"

 required

 />
```

```
</FormField>
```

```
<FormField label={t.exhibitorNumber} required>
```

```
<input
```

```
 type="text"
```

```
 value={formData.exhibitorNumber}
```

```
 onChange={(e) => handleInputChange('exhibitorNumber', e.target.value)}
```

```
 placeholder={t.exhibitorPlaceholder}
```

```
 className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-green-500 focus:border-green-500 transition-all"
```

```
 required
```

```
</FormField>
```

```
</div>
```

```
<div className="grid grid-cols-1 md:grid-cols-2 gap-6 mt-6">
```

```
<FormField label={t.period} required>
```

```
<select
```

```
 value={formData.period}
```

```
 onChange={(e) => handleInputChange('period', e.target.value)}
```

```
 className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-green-500 focus:border-green-500 transition-all appearance-none bg-white"
```

```
 required
```

```
 <option value="">{t.periodPlaceholder}</option>
```

```
 <option value="first">{t.firstPeriod}</option>
```

```
 <option value="second">{t.secondPeriod}</option>
```

```
</select>
```

```
</FormField>
```

```
<FormField label={t.seatNumber} required>
```

```
<div className="relative">
```

```
<MapPin className="absolute left-3 top-3 w-5 h-5 text-green-500" />
```

```
<input
```

```
 type="text"
```

```
 value={formData.seatNumber}
```

```
 onChange={(e) => handleInputChange('seatNumber', e.target.value)}
```

```
 placeholder={t.seatPlaceholder}
```

```
 className="w-full pl-12 pr-4 py-3 border border-gray-300 rounded-lg focus:ring-2
focus:ring-green-500 focus:border-green-500 transition-all"
```

```
 required
```

```
</div>
```

```
</FormField>
```

```
</div>
```

```
</div>
```

```
{/* Order Details */}
```

```
<div className="bg-white rounded-xl shadow-sm p-6 border border-green-100">
```

```
<div className="flex items-center gap-3 mb-6">
```

```
<Flower2 className="w-6 h-6 text-green-600" />
```

```
<h3 className="text-lg font-semibold text-gray-800">{t.orderDetails}</h3>
```

```
</div>
```

```
<div className="space-y-6">
```

```
<FormField label={t.flowerType} required>
```

```
 <textarea
```

```
 value={formData.flowerType}
```

```
 onChange={(e) => handleInputChange('flowerType', e.target.value)}
```

```
 placeholder={t.flowerPlaceholder}
```

```
 rows={4}
```

```
 className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-green-500 focus:border-green-500 transition-all resize-none"
```

```
 required
```

```
 />
```

```
</FormField>
```

```
<div className="grid grid-cols-1 md:grid-cols-2 gap-6">
```

```
 <FormField label={t.quantity} required>
```

```
 <input
```

```
 type="text"
```

```
 value={formData.quantity}
```

```
 onChange={(e) => handleInputChange('quantity', e.target.value)}
```

```
 placeholder={t.quantityPlaceholder}
```

```
 className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-green-500 focus:border-green-500 transition-all"
```

```
 required
```

```
 />
```

```
 </FormField>
```

```
<FormField label={t.colors}>
```

```
 <div className="relative">
```

```
 <Palette className="absolute left-3 top-3 w-5 h-5 text-green-500" />
```

```
<input
 type="text"
 value={formData.colors}
 onChange={(e) => handleInputChange('colors', e.target.value)}
 placeholder={t.colorsPlaceholder}
 className="w-full pl-12 pr-4 py-3 border border-gray-300 rounded-lg focus:ring-2
focus:ring-green-500 focus:border-green-500 transition-all"
/>
</div>
</FormField>
</div>
```

```
<FormField label={t.budget}>
 <div className="relative">
 <DollarSign className="absolute left-3 top-3 w-5 h-5 text-green-500" />
 <input
 type="text"
 value={formData.budget}
 onChange={(e) => handleInputChange('budget', e.target.value)}
 placeholder={t.budgetPlaceholder}
 className="w-full pl-12 pr-4 py-3 border border-gray-300 rounded-lg focus:ring-2
focus:ring-green-500 focus:border-green-500 transition-all"
 />
 </div>
</FormField>
```

```
<div className="grid grid-cols-1 md:grid-cols-2 gap-6">
 <FormField label={t.deliveryDate} required>
```

```
<div className="relative">

 <Calendar className="absolute left-3 top-3 w-5 h-5 text-green-500" />

 <input

 type="date"

 value={formData.deliveryDate}

 onChange={(e) => handleInputChange('deliveryDate', e.target.value)}

 className="w-full pl-12 pr-4 py-3 border border-gray-300 rounded-lg focus:ring-2
focus:ring-green-500 focus:border-green-500 transition-all"

 required

 />

</div>

</FormField>
```

```
<FormField label={t.deliveryTime}>

 <div className="relative">

 <Clock className="absolute left-3 top-3 w-5 h-5 text-green-500" />

 <select

 value={formData.deliveryTime}

 onChange={(e) => handleInputChange('deliveryTime', e.target.value)}

 className="w-full pl-12 pr-4 py-3 border border-gray-300 rounded-lg focus:ring-2
focus:ring-green-500 focus:border-green-500 transition-all appearance-none bg-white"

 >

 <option value="">{language === 'en' ? 'Select time' : '時間を選択'}</option>

 <option value="morning">{t.morning}</option>

 <option value="afternoon">{t.afternoon}</option>

 <option value="evening">{t.evening}</option>

 </select>

 </div>
```

```
</FormField>
```

```
</div>
```

```
<FormField label={t.specialRequests}>
```

```
 <div className="relative">
```

```
 <FileText className="absolute left-3 top-3 w-5 h-5 text-green-500" />
```

```
 <textarea
```

```
 value={formData.specialRequests}
```

```
 onChange={(e) => handleInputChange('specialRequests', e.target.value)}
```

```
 placeholder={t.specialPlaceholder}
```

```
 rows={4}
```

```
 className="w-full pl-12 pr-4 py-3 border border-gray-300 rounded-lg focus:ring-2
focus:ring-green-500 focus:border-green-500 transition-all resize-none"
```

```
 />
```

```
 </div>
```

```
</FormField>
```

```
</div>
```

```
</div>
```

```
{/* Image Upload */}
```

```
<div className="bg-white rounded-xl shadow-sm p-6 border border-green-100">
```

```
 <ImageUpload
```

```
 onImageChange={setImages}
```

```
 label={t.sketchUpload}
```

```
 maxFiles={3}
```

```
 />
```

```
</div>
```



```
{/* Submit Button */}
```

```
<div className="flex justify-center pt-4">
```

```
<button
```

```
type="submit"
```

```
className="bg-gradient-to-r from-green-600 to-emerald-600 text-white font-semibold px-8
py-4 rounded-lg hover:from-green-700 hover:to-emerald-700 transform hover:scale-105 transition-
all duration-200 shadow-lg hover:shadow-xl flex items-center gap-3"
```

```
>
```

```
<Mail className="w-5 h-5" />
```

```
{t.submitButton}
```

```
</button>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
{/* Footer */}
```

```
<div className="bg-white border-t border-green-100 mt-12">
```

```
<div className="max-w-4xl mx-auto px-4 py-6 text-center">
```

```
<p className="text-sm text-gray-600">
```

```
© 2025 {t.title} | Powered by Bloomwinkel
```

```
</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
);
```

```
}
```

```
export default App;
```

```
```
```

```
---
```

```
## src/translations.ts
```

```
```typescript
```

```
export const translations = {
```

```
 en: {
```

```
 title: "The 106th Sogetsu Annual Exhibition",
```

```
 subtitle: "Flower Material Order Form",
```

```
 contact: "To: DeBloemenwinkel",
```

```
 email: "E-mail: deblwinkel@gmail.com",
```

```
 deadline: "Please fill in the following information in detail and submit it with your sketch to the
above email address by Thursday, September 18th.",
```

```
 payment: "The billing amount will be communicated to you by the florist \"Bloomwinkel\". You can
pay by credit card.",
```

```
 personalInfo: "Personal Information",
```

```
 orderDetails: "Order Details",
```

```
 name: "Full Name",
```

```
 namePlaceholder: "Enter your full name",
```

```
 phone: "Phone Number",
```

```
 phonePlaceholder: "Enter your phone number",
```

```
 emailField: "Email Address",
```

```
 emailPlaceholder: "Enter your email address",
```

```
 exhibitorNumber: "Exhibitor Number / ID",
```

```
 exhibitorPlaceholder: "Enter your exhibitor number",
```

```
period: "Exhibition Period",
periodPlaceholder: "Select exhibition period",
firstPeriod: "First Period",
secondPeriod: "Second Period",
seatNumber: "Seat Number",
seatPlaceholder: "Enter your seat number",
flowerType: "Flower Type / Materials",
flowerPlaceholder: "Describe the flowers and materials needed",
quantity: "Quantity",
quantityPlaceholder: "Specify quantity needed",
colors: "Preferred Colors",
colorsPlaceholder: "Specify color preferences",
budget: "Budget Range (¥)",
budgetPlaceholder: "Enter your budget range",
deliveryDate: "Preferred Delivery Date",
deliveryTime: "Preferred Delivery Time",
specialRequests: "Special Requirements / Notes",
specialPlaceholder: "Any special requirements or additional notes",
sketchUpload: "Upload Sketch / Reference Images",
submitButton: "Submit Order",
required: "Required",
optional: "Optional",
morning: "Morning (9:00-12:00)",
afternoon: "Afternoon (12:00-17:00)",
evening: "Evening (17:00-20:00)"
},
ja: {
```

title: "第106回草月いけばな展",

subtitle: "花材注文書",

contact: "宛先：DeBloemenwinkel ブルームウィンクル",

email: "E-mail: deblwinkel@gmail.com",

deadline: "下記の情報を詳細に記入し、スケッチと共に9月18日（木）までに上記メールアドレスへ提出してください。",

payment: "請求金額は花屋「Bloomwinkel」よりお知らせいたします。クレジットカードでお支払いいただけます。",

personalInfo: "個人情報",

orderDetails: "注文詳細",

name: "お名前",

namePlaceholder: "お名前をご記入ください",

phone: "電話番号",

phonePlaceholder: "電話番号をご記入ください",

emailField: "メールアドレス",

emailPlaceholder: "メールアドレスをご記入ください",

exhibitorNumber: "出品者番号 / ID",

exhibitorPlaceholder: "出品者番号をご記入ください",

period: "展示期間",

periodPlaceholder: "展示期間を選択してください",

firstPeriod: "前期",

secondPeriod: "後期",

seatNumber: "花席No.",

seatPlaceholder: "花席番号をご記入ください",

flowerType: "花材の種類",

flowerPlaceholder: "必要な花材について詳しくご記入ください",

quantity: "数量",

quantityPlaceholder: "必要な数量をご記入ください",

colors: "希望する色",

colorsPlaceholder: "色の希望をご記入ください",

budget: "予算範囲 (¥) ",

budgetPlaceholder: "予算範囲をご記入ください",

deliveryDate: "希望配送日",

deliveryTime: "希望配送時間",

specialRequests: "特別な要望 / 備考",

specialPlaceholder: "特別な要望や追加の備考があればご記入ください",

sketchUpload: "スケッチ / 参考画像のアップロード",

submitButton: "注文を送信",

required: "必須",

optional: "任意",

morning: "午前 (9:00-12:00) ",

afternoon: "午後 (12:00-17:00) ",

evening: "夕方 (17:00-20:00) "

}

};

...

---

```
src/components/FormField.tsx
```

```
```tsx
```

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

```
interface FormFieldProps {
```

```
  label: string;
```

```
  required?: boolean;
```

```
  children: React.ReactNode;
```

```
}
```

```
export default function FormField({ label, required = false, children }: FormFieldProps) {
```

```
  return (
```

```
    <div className="space-y-2">
```

```
      <label className="block text-sm font-semibold text-gray-700">
```

```
        {label}
```

```
        {required && <span className="text-red-500 ml-1">*</span>}
```

```
      </label>
```

```
      {children}
```

```
    </div>
```

```
  );
```

```
}
```

```
```
```

---

```
src/components/ImageUpload.tsx
```

```
``tsx
```

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

```
import { Upload, X, Image as ImageIcon } from 'lucide-react';
```

```
interface ImageUploadProps {
```

```
 onImageChange: (files: File[]) => void;
```

```
 label: string;
```

```
 maxFiles?: number;
```

```
}
```

```
export default function ImageUpload({ onImageChange, label, maxFiles = 3 }: ImageUploadProps) {
```

```
 const [images, setImages] = useState<File[]>([]);
```

```
 const [previews, setPreviews] = useState<string[]>([]);
```

```
 const fileInputRef = useRef<HTMLInputElement>(null);
```

```
 const handleFileSelect = (event: React.ChangeEvent<HTMLInputElement>) => {
```

```
 const files = Array.from(event.target.files || []);
```

```
 const validFiles = files.filter(file => file.type.startsWith('image/'));
```

```
 if (images.length + validFiles.length > maxFiles) {
```

```
 alert(`Maximum ${maxFiles} images allowed`);
```

```
 return;
```

```
 }
```

```

const newImages = [...images, ...validFiles];

setImages(newImages);

onImageChange(newImages);

// Create previews

validFiles.forEach(file => {

 const reader = new FileReader();

 reader.onload = (e) => {

 setPreviews(prev => [...prev, e.target?.result as string]);

 };

 reader.readAsDataURL(file);

});

};

const removeImage = (index: number) => {

 const newImages = images.filter((_, i) => i !== index);

 const newPreviews = previews.filter((_, i) => i !== index);

 setImages(newImages);

 setPreviews(newPreviews);

 onImageChange(newImages);

};

return (

 <div className="space-y-4">

 <label className="block text-sm font-semibold text-gray-700 mb-2">

 {label}

```



```
</label>
```

```
<div
```

```
 onClick={() => fileInputRef.current?.click()}
```

```
 className="border-2 border-dashed border-green-300 rounded-lg p-6 text-center cursor-
pointer hover:border-green-400 hover:bg-green-50 transition-all"
```

```
>
```

```
 <Upload className="w-8 h-8 text-green-500 mx-auto mb-2" />
```

```
 <p className="text-sm text-gray-600 mb-1">Click to upload sketches/images</p>
```

```
 <p className="text-xs text-gray-500">PNG, JPG up to 10MB (max {maxFiles} files)</p>
```

```
</div>
```

```
<input
```

```
 ref={fileInputRef}
```

```
 type="file"
```

```
 multiple
```

```
 accept="image/*"
```

```
 onChange={handleFileSelect}
```

```
 className="hidden"
```

```
/>
```

```
{previews.length > 0 && (
```

```
 <div className="grid grid-cols-2 sm:grid-cols-3 gap-4">
```

```
 {previews.map((preview, index) => (
```

```
 <div key={index} className="relative group">
```

```
 <img
```

```
 src={preview}
```

```

 alt={`Preview ${index + 1}`}

 className="w-full h-24 sm:h-32 object-cover rounded-lg shadow-sm"

 />

 <button

 onClick={() => removeImage(index)}

 className="absolute -top-2 -right-2 bg-red-500 text-white rounded-full p-1 opacity-0
group-hover:opacity-100 transition-opacity"

 >

 <X className="w-4 h-4" />

 </button>

 </div>

)})

</div>

)}

</div>

);

}

...

```

```

src/components/LanguageToggle.tsx

```

```

````tsx

```

```

import React from 'react';

```

```

import { Globe } from 'lucide-react';

```

```
interface LanguageToggleProps {
  language: 'en' | 'ja';
  onToggle: (lang: 'en' | 'ja') => void;
}
```

```
export default function LanguageToggle({ language, onToggle }: LanguageToggleProps) {
  return (
    <div className="flex items-center gap-2 bg-white rounded-full p-2 shadow-md">
      <Globe className="w-4 h-4 text-green-600" />
      <button
        onClick={() => onToggle('en')}
        className={`px-3 py-1 rounded-full text-sm font-medium transition-all ${
          language === 'en'
            ? 'bg-green-600 text-white shadow-sm'
            : 'text-gray-600 hover:text-green-600'
        }`}
      >
        EN
      </button>
      <button
        onClick={() => onToggle('ja')}
        className={`px-3 py-1 rounded-full text-sm font-medium transition-all ${
          language === 'ja'
            ? 'bg-green-600 text-white shadow-sm'
            : 'text-gray-600 hover:text-green-600'
        }`}
      >
```

日本語

</button>

</div>

);

}

...

src/vite-env.d.ts

``typescript

/// <reference types="vite/client" />

...

vite.config.ts

``typescript

import { defineConfig } from 'vite';

import react from '@vitejs/plugin-react';

// <https://vitejs.dev/config/>

export default defineConfig({

plugins: [react()],

optimizeDeps: {

```
    exclude: ['lucide-react'],  
  },  
});  
```
```

---

## tsconfig.json

```
``json
{
 "files": [],
 "references": [
 { "path": "./tsconfig.app.json" },
 { "path": "./tsconfig.node.json" }
]
}
```
```

tsconfig.app.json

```
``json  
{  
  "compilerOptions": {  
    "target": "ES2020",
```

```
"useDefineForClassFields": true,  
"lib": ["ES2020", "DOM", "DOM.Iterable"],  
"module": "ESNext",  
"skipLibCheck": true,
```

```
/* Bundler mode */
```

```
"moduleResolution": "bundler",  
"allowImportingTsExtensions": true,  
"isolatedModules": true,  
"moduleDetection": "force",  
"noEmit": true,  
"jsx": "react-jsx",
```

```
/* Linting */
```

```
"strict": true,  
"noUnusedLocals": true,  
"noUnusedParameters": true,  
"noFallthroughCasesInSwitch": true
```

```
},
```

```
"include": ["src"]
```

```
}
```

```
...
```

```
---
```

```
## tsconfig.node.json
```

```
```json
{
 "compilerOptions": {
 "target": "ES2022",
 "lib": ["ES2023"],
 "module": "ESNext",
 "skipLibCheck": true,

 /* Bundler mode */

 "moduleResolution": "bundler",
 "allowImportingTsExtensions": true,
 "isolatedModules": true,
 "moduleDetection": "force",
 "noEmit": true,

 /* Linting */

 "strict": true,
 "noUnusedLocals": true,
 "noUnusedParameters": true,
 "noFallthroughCasesInSwitch": true
 },
 "include": ["vite.config.ts"]
}
```

---
```

```
## tailwind.config.js
```

```
````javascript
```

```
/** @type {import('tailwindcss').Config} */
```

```
export default {
```

```
 content: ['./index.html', './src/**/*.{js,ts,jsx,tsx}'],
```

```
 theme: {
```

```
 extend: {},
```

```
 },
```

```
 plugins: [],
```

```
};
```

```
````
```

```
---
```

```
## postcss.config.js
```

```
````javascript
```

```
export default {
```

```
 plugins: {
```

```
 tailwindcss: {},
```

```
 autoprefixer: {},
```

```
 },
```

```
};
```

```
````
```

```
---
```



```
## eslint.config.js
```

```
````javascript
```

```
import js from '@eslint/js';
```

```
import globals from 'globals';
```

```
import reactHooks from 'eslint-plugin-react-hooks';
```

```
import reactRefresh from 'eslint-plugin-react-refresh';
```

```
import tseslint from 'typescript-eslint';
```

```
export default tseslint.config(
 { ignores: ['dist'] },
 {
 extends: [js.configs.recommended, ...tseslint.configs.recommended],
 files: ['**/*.ts', '**/*.tsx'],
 languageOptions: {
 ecmaVersion: 2020,
 globals: globals.browser,
 },
 plugins: {
 'react-hooks': reactHooks,
 'react-refresh': reactRefresh,
 },
 rules: {
 ...reactHooks.configs.recommended.rules,
 'react-refresh/only-export-components': [
 'warn',
],
 },
 },
]);
```

```
{ allowConstantExport: true },

],

},

}

);
...

```

## ## セットアップ手順 / Setup Instructions

### 1. \*\*新しいフォルダを作成\*\*

```
``bash

mkdir flowerorderform

cd flowerorderform
...
```

### 2. \*\*上記のファイルをそれぞれ作成\*\*

- 各ファイルの内容をコピー＆ペーストしてください

### 3. \*\*依存関係をインストール\*\*

```
``bash

npm install
...
```

### 4. \*\*開発サーバーを起動\*\*

```
```bash
```

```
npm run dev
```

```
```
```

## 5. \*\*本番用ビルド\*\*

```
```bash
```

```
npm run build
```

```
```
```

## ## 使用方法 / Usage

- ブラウザで `http://localhost:5173` にアクセス
- 言語切り替えボタンで英語/日本語を選択
- フォームに必要事項を入力
- 画像をアップロード（最大3ファイル）
- 送信ボタンをクリック

---

**\*\*注意\*\*:** このファイルをデスクトップに保存して、必要な部分をコピー&ペーストしてプロジェクトを作成してください。