напиши компоненту и подключи ее в app.tsx которая будет публиковать геометрические задачи компонента предусматривает описание задание задачи и ее чертеж

компонент GeometryTask.tsx

import React, { useState } from "react";

interface GeometryTaskProps {

onSubmit: (task: { description: string; drawing: string }) => void; // Callback для отправки задачи

}

const GeometryTask: React.FC<GeometryTaskProps> = ({ onSubmit }) => {

const [description, setDescription] = useState<string>(""); // Описание задачи

const [drawing, setDrawing] = useState<string | null>(null); // URL чертежа

// Обработка загрузки файла (чертеж)

const handleFileUpload = (e: React.ChangeEvent<HTMLInputElement>) => {

if (e.target.files && e.target.files[0]) {

const file = e.target.files[0];

const reader = new FileReader();

reader.onload = () => {

if (typeof reader.result === "string") {

setDrawing(reader.result); // Сохраняем base64 представление изображения

}

};

reader.readAsDataURL(file);

}

};

// Отправка задачи

const handleSubmit = () => {

if (!description.trim()) {

alert("Введите описание задачи!");

return;

}

if (!drawing) {

alert("Добавьте чертеж!");

return;

}

onSubmit({ description, drawing }); // Вызываем callback для отправки задачи

setDescription(""); // Сбрасываем форму

setDrawing(null);

};

return (

<div style={{ marginBottom: "20px" }}>

<h2>Создание геометрической задачи</h2>

<textarea

value={description}

onChange={(e) => setDescription(e.target.value)}

placeholder="Введите описание задачи..."

style={{ width: "100%", height: "100px", padding: "8px", marginBottom: "10px" }}

/>

<input

type="file"

accept="image/\*"

onChange={handleFileUpload}

style={{ marginBottom: "10px" }}

/>

{drawing && (

<img

src={drawing}

alt="Чертеж"

style={{ maxWidth: "300px", marginBottom: "10px" }}

/>

)}

<button onClick={handleSubmit} style={{ padding: "8px 16px" }}>

Опубликовать задачу

</button>

</div>

);

};

export default GeometryTask;

компонент

**Шаг 2: Подключение компонента в App.tsx**

Теперь добавим компонент **GeometryTask** в основной файл **App.tsx**.

import React, { useState } from "react";

import ImageUpload from "./components/ImageUpload";

import ImageList from "./components/ImageList";

import MathSolver from "./components/MathSolver";

import GeometryTask from "./components/GeometryTask"; // Подключаем новый компонент

const App: React.FC = () => {

const [images, setImages] = useState<{ id: string; url: string; tags: string[] }[]>([]);

const [geometryTasks, setGeometryTasks] = useState<

{ description: string; drawing: string }[]

>([]);

const handleImageUpload = (file: File) => {

const imageUrl = URL.createObjectURL(file);

setImages((prevImages) => [

...prevImages,

{ id: file.name, url: imageUrl, tags: [] },

]);

};

const handleTagsChange = (id: string, tags: string[]) => {

setImages((prevImages) =>

prevImages.map((image) =>

image.id === id ? { ...image, tags } : image

)

);

};

// Обработка отправки математического выражения

const handleMathExpression = (expression: string) => {

console.log("Отправляем выражение на сервер:", expression);

};

// Обработка отправки геометрической задачи

const handleGeometryTaskSubmit = (task: { description: string; drawing: string }) => {

console.log("Новая геометрическая задача:", task);

setGeometryTasks((prevTasks) => [...prevTasks, task]);

};

return (

<div>

<h1>Менеджер изображений, математический решатель и геометрические задачи</h1>

<MathSolver onSolve={handleMathExpression} />

<GeometryTask onSubmit={handleGeometryTaskSubmit} /> {/\* Добавляем компонент \*/}

<ImageUpload onUpload={handleImageUpload} />

<ImageList images={images} onTagsChange={handleTagsChange} />

{/\* Отображение опубликованных геометрических задач \*/}

<h2>Опубликованные геометрические задачи:</h2>

{geometryTasks.length === 0 && <p>Задач еще нет.</p>}

{geometryTasks.map((task, index) => (

<div key={index} style={{ marginBottom: "20px", border: "1px solid #ccc", padding: "10px" }}>

<h3>Задача #{index + 1}</h3>

<p>{task.description}</p>

<img

src={task.drawing}

alt="Чертеж"

style={{ maxWidth: "300px", marginTop: "10px" }}

/>

</div>

))}

</div>

);

};

export default App;