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

import { useEffect } from 'react';



const HandwrittenDigitPredictionApp = () => {

const [drawing, setDrawing] = useState(false);

const [lastPos, setLastPos] = useState({ x: 0, y: 0 });

const [imageData, setImageData] = useState('');

const [prediction, setPrediction] = useState('');



const handleMouseDown = (e: React.MouseEvent<HTMLCanvasElement>) => {

  const canvas = e.target as HTMLCanvasElement;
  const rect = canvas.getBoundingClientRect();

  const x = e.clientX - rect.left;

  const y = e.clientY - rect.top;

  setDrawing(true);

  setLastPos({ x, y });

};



const handleMouseMove = (e: React.MouseEvent<HTMLCanvasElement>) => {

  if (!drawing) return;

  const canvas = e.target as HTMLCanvasElement;

  const rect = canvas.getBoundingClientRect();
  const x = e.clientX - rect.left;

  const y = e.clientY - rect.top;

  const ctx = canvas.getContext('2d');

  if (!ctx) return;

  ctx.beginPath();

  ctx.lineWidth = 10;

  ctx.lineCap = 'round';

  ctx.lineJoin = 'round';

  ctx.moveTo(lastPos.x, lastPos.y);

  ctx.lineTo(x, y);

  ctx.stroke();
```

```
  setLastPos({ x, y });
};


const handleMouseUp = () => {
  setDrawing(false);
  const canvas = document.getElementById('canvas') as HTMLCanvasElement;
  if (!canvas) return;
  const ctx = canvas.getContext('2d');
  if (!ctx) return;
  setImageData(canvas.toDataURL());
};
```

```
const handlePredict = () => {

  // TO DO: implement prediction logic here

  // For now, just set a random prediction

  setPrediction(Math.floor(Math.random() * 10).toString());

};



useEffect(() => {

  const canvas = document.getElementById('canvas') as HTMLCanvasElement;

  if (!canvas) return;

  const ctx = canvas.getContext('2d');
  if (!ctx) return;

  ctx.clearRect(0, 0, canvas.width, canvas.height);

}, [imageData]);



return (

  <div className="max-w-md mx-auto p-4 pt-6 md:p-6 lg:p-12 xl:p-24">

    <h1 className="text-3xl text-gray-900 leading-tight">Handwritten Digit Prediction App</h1>

    <div className="flex flex-col items-center justify-center mt-6">

      <canvas

        id="canvas"

        width={280}
        height={280}

        className="border-2 border-gray-200 rounded-lg"

        onMouseDown={handleMouseDown}

        onMouseMove={handleMouseMove}

        onMouseUp={handleMouseUp}

      />

      <button

        className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded mt-4"

        onClick={handlePredict}

      >

        Predict
```

```
      </button>

      <p className="text-lg text-gray-600 mt-4">Prediction: {prediction}</p>

    </div>

  </div>

);

};



export default HandwrittenDigitPredictionApp
```