

Snippet di codice

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

```
// Componente Header per la navigazione
```

```
const Header = ({ currentPage, onNavigate }) => {
```

```
  return (
```

```
    <header className="bg-gradient-to-r from-blue-600 to-blue-800 text-white p-4 shadow-lg rounded-b-lg">
```

```
      <div className="container mx-auto flex justify-between items-center">
```

```
        <h1 className="text-3xl font-bold text-white tracking-wide">SyDiDOa</h1>
```

```
        <nav>
```

```
          <ul className="flex space-x-6">
```

```
            <li>
```

```
              <button
```

```
                onClick={() => onNavigate('home')}
```

```
                className={`px-4 py-2 rounded-full transition-all duration-300 ${
```

```
                  currentPage === 'home' ? 'bg-white text-blue-800 shadow-md' : 'text-white hover:bg-blue-
```

```
700'
```

```
                }`
```

```
              >
```

```
                Home
```

```
              </button>
```

```
            </li>
```

```
            <li>
```

```
              <button
```

```
                onClick={() => onNavigate('profile')}
```

```
                className={`px-4 py-2 rounded-full transition-all duration-300 ${
```

```
                  currentPage === 'profile' ? 'bg-white text-blue-800 shadow-md' : 'text-white hover:bg-blue-
```

```
700'
```

```
                }`
```

```
              >
```

```
                Profilo
```

```
              </button>
```

```
            </li>
```

```
            <li>
```

```
              <button
```

```
                onClick={() => onNavigate('consent')}
```

```
                className={`px-4 py-2 rounded-full transition-all duration-300 ${
```

```
                  currentPage === 'consent' ? 'bg-white text-blue-800 shadow-md' : 'text-white hover:bg-blue-
```

```
700'
```

```
                }`
```

```
              >
```

```
                Consenso
```

```
              </button>
```

```
            </li>
```

```
            <li>
```

```
              <button
```

```
                onClick={() => onNavigate('imageUpload')}
```

```
                className={`px-4 py-2 rounded-full transition-all duration-300 ${
```

```
                  currentPage === 'imageUpload' ? 'bg-white text-blue-800 shadow-md' : 'text-white hover:bg-
```

```
blue-700'
```

```
                }`
```

```
              >
```

```
                Immagini
```

```
              </button>
```

```
            </li>
```

```
            <li>
```

```
              <button
```

```
                onClick={() => onNavigate('questionnaire')}
```

```

        className={`px-4 py-2 rounded-full transition-all duration-300 ${
          currentPage === 'questionnaire' ? 'bg-white text-blue-800 shadow-md' : 'text-white hover:bg-
blue-700'
        }}
      >
        Questionari
      </button>
    </li>
    <li>
      <button
        onClick={() => onNavigate('therapyPlan')}
        className={`px-4 py-2 rounded-full transition-all duration-300 ${
          currentPage === 'therapyPlan' ? 'bg-white text-blue-800 shadow-md' : 'text-white hover:bg-
blue-700'
        }}
      >
        Piano Terapeutico
      </button>
    </li>
  </ul>
</nav>
</div>
</header>
);
};

```

// Componente Footer

```

const Footer = () => {
  return (
    <footer className="bg-gray-800 text-white p-4 mt-8 rounded-t-lg shadow-inner">
      <div className="container mx-auto text-center text-sm">
        <p>&copy; {new Date().getFullYear()} SyDiDOa Project. Tutti i diritti riservati.</p>
        <p className="mt-2">Finanziato dall'Unione Europea - NextGenerationEU</p>
      </div>
    </footer>
  );
};

```

// Componente Home Page

```

const Home = () => {
  return (
    <div className="flex flex-col items-center justify-center p-8 bg-white rounded-lg shadow-xl max-w-
4xl mx-auto my-8">
      <h2 className="text-4xl font-extrabold text-blue-800 mb-6 text-center">Benvenuti in
SyDiDOa</h2>
      <p className="text-lg text-gray-700 text-center leading-relaxed">
        Il progetto SyDiDOa - Symptomatic Digital Drugs for Osteoarthritis mira a rivoluzionare il
trattamento dell'osteoartrosi del ginocchio attraverso lo sviluppo di un innovativo farmaco digitale
(Digital Drug) basato su web-app.
      </p>
      <p className="text-lg text-gray-700 text-center leading-relaxed mt-4">
        Siamo impegnati a migliorare la qualità della vita dei pazienti, fornendo soluzioni terapeutiche
personalizzate, accessibili e basate su evidenze scientifiche, sfruttando il potenziale delle tecnologie
digitali e dell'intelligenza artificiale.
      </p>
      <div className="mt-8 grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6 w-full">
        <div className="bg-blue-50 p-6 rounded-xl shadow-md flex flex-col items-center text-center
transform hover:scale-105 transition-transform duration-300">
          <span className="text-5xl mb-3">💡 </span>

```

```

        <h3 className="text-xl font-semibold text-blue-700 mb-2">Innovazione</h3>
        <p className="text-gray-600">Sviluppo del primo farmaco digitale per l'osteoartrosi.</p>
    </div>
    <div className="bg-green-50 p-6 rounded-xl shadow-md flex flex-col items-center text-center
transform hover:scale-105 transition-transform duration-300">
        <span className="text-5xl mb-3">🌐</span>
        <h3 className="text-xl font-semibold text-green-700 mb-2">Accessibilità</h3>
        <p className="text-gray-600">Una web-app per una terapia comoda e sempre disponibile.</p>
    </div>
    <div className="bg-purple-50 p-6 rounded-xl shadow-md flex flex-col items-center text-center
transform hover:scale-105 transition-transform duration-300">
        <span className="text-5xl mb-3">📊</span>
        <h3 className="text-xl font-semibold text-purple-700 mb-2">Evidenza</h3>
        <p className="text-gray-600">Terapie basate su rigorose ricerche scientifiche e studi
        clinici.</p>
    </div>
</div>
);
};

// Componente Profilo Utente
const Profile = ({ userData, setUserData }) => {
    const handleChange = (e) => {
        const { name, value } = e.target;
        setUserData((prevData) => ({ ...prevData, [name]: value }));
    };

    const handleSubmit = (e) => {
        e.preventDefault();
        // Qui si invierebbero i dati al backend
        console.log('Dati profilo salvati:', userData);
        alert('Dati profilo aggiornati con successo!');
    };

    return (
        <div className="p-8 bg-white rounded-lg shadow-xl max-w-3xl mx-auto my-8">
            <h2 className="text-3xl font-bold text-blue-800 mb-6 text-center">Il tuo Profilo</h2>
            <form onSubmit={handleSubmit} className="space-y-6">
                <div>
                    <label htmlFor="name" className="block text-lg font-medium text-gray-700 mb-
2">Nome:</label>
                    <input
                        type="text"
                        id="name"
                        name="name"
                        value={userData.name || ""}
                        onChange={handleChange}
                        className="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500 sm:text-lg"
                        placeholder="Il tuo nome"
                    />
                </div>
                <div>
                    <label htmlFor="surname" className="block text-lg font-medium text-gray-700 mb-
2">Cognome:</label>
                    <input
                        type="text"

```

```

        id="surname"
        name="surname"
        value={userData.surname || ""}
        onChange={handleChange}
        className="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500 sm:text-lg"
        placeholder="Il tuo cognome"
      />
    </div>
    <div>
      <label htmlFor="dob" className="block text-lg font-medium text-gray-700 mb-2">Data di
Nascita:</label>
      <input
        type="date"
        id="dob"
        name="dob"
        value={userData.dob || ""}
        onChange={handleChange}
        className="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500 sm:text-lg"
      />
    </div>
    <div>
      <label htmlFor="height" className="block text-lg font-medium text-gray-700 mb-2">Altezza
(cm):</label>
      <input
        type="number"
        id="height"
        name="height"
        value={userData.height || ""}
        onChange={handleChange}
        className="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500 sm:text-lg"
        placeholder="Es. 175"
      />
    </div>
    <div>
      <label htmlFor="weight" className="block text-lg font-medium text-gray-700 mb-2">Peso
(kg):</label>
      <input
        type="number"
        id="weight"
        name="weight"
        value={userData.weight || ""}
        onChange={handleChange}
        className="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500 sm:text-lg"
        placeholder="Es. 70"
      />
    </div>
    <button
      type="submit"
      className="w-full bg-blue-600 hover:bg-blue-700 text-white font-bold py-3 px-6 rounded-full
shadow-lg transition-all duration-300 transform hover:scale-105"
    >
      Salva Profilo
    </button>
  </form>
</div>

```

```

    );
  };

  // Componente Consenso
  const ConsentForm = ({ consentGiven, setConsentGiven }) => {
    const handleConsentChange = (e) => {
      setConsentGiven(e.target.checked);
    };

    const handleSubmit = (e) => {
      e.preventDefault();
      if (consentGiven) {
        // Qui si invierebbe lo stato del consenso al backend
        console.log('Consenso accettato. ');
        alert('Consenso accettato con successo!');
      } else {
        alert('È necessario accettare il consenso per procedere. ');
      }
    };

    return (
      <div className="p-8 bg-white rounded-lg shadow-xl max-w-3xl mx-auto my-8">
        <h2 className="text-3xl font-bold text-blue-800 mb-6 text-center">Gestione del Consenso</h2>
        <div className="prose max-w-none text-gray-700 text-lg leading-relaxed mb-6">
          <p>
            Il progetto SyDiDOa si impegna a proteggere la tua privacy e i tuoi dati personali. Per poter
            utilizzare la web-app e partecipare alla ricerca, è necessario fornire il tuo consenso informato al
            trattamento dei dati.
          </p>
          <h3 className="text-2xl font-semibold text-blue-700 mt-6 mb-3">Informativa sulla Privacy e
            Trattamento Dati:</h3>
          <p>
            I dati raccolti (informazioni personali, dati antropometrici, immagini radiografiche, risposte ai
            questionari, dati di attività) saranno utilizzati esclusivamente per le finalità di ricerca del progetto
            SyDiDOa, per migliorare il trattamento dell'osteoartrosi del ginocchio e per lo sviluppo del dispositivo
            medico digitale. I tuoi dati saranno trattati in forma anonima o pseudonimizzata ove possibile e in
            conformità con il Regolamento Generale sulla Protezione dei Dati (GDPR 2016/679).
          </p>
          <p>
            Hai il diritto di accedere ai tuoi dati, richiederne la rettifica o la cancellazione, limitare il
            trattamento, opporsi al trattamento e alla portabilità dei dati. Per maggiori dettagli, consulta la nostra
            informativa completa.
          </p>
          <h3 className="text-2xl font-semibold text-blue-700 mt-6 mb-3">Consenso Informato al
            Trattamento:</h3>
          <p>
            Comprendo le finalità e le modalità di trattamento dei miei dati personali e di salute, come
            descritto nell'informativa. Dichiaro di aver letto e compreso l'informativa sulla privacy e di acconsentire
            al trattamento dei miei dati per le finalità specificate nel progetto SyDiDOa.
          </p>
        </div>
        <form onSubmit={handleSubmit}>
          <div className="flex items-center mb-6">
            <input
              type="checkbox"
              id="consentCheckbox"
              checked={consentGiven}
              onChange={handleConsentChange}
              className="h-6 w-6 text-blue-600 rounded-md border-gray-300 focus:ring-blue-500"
            />

```

```

    />
    <label htmlFor="consentCheckbox" className="ml-3 text-lg text-gray-800 font-medium">
      Acconsento al trattamento dei miei dati personali per le finalità del progetto SyDiDOa.
    </label>
  </div>
  <button
    type="submit"
    className="w-full bg-blue-600 hover:bg-blue-700 text-white font-bold py-3 px-6 rounded-full
shadow-lg transition-all duration-300 transform hover:scale-105"
    disabled={!consentGiven}
  >
    Conferma Consenso
  </button>
</form>
</div>
);
};

```

// Componente Caricamento Immagini Radiografiche

```

const ImageUpload = ({ uploadedImages, setUploadedImages }) => {
  const handleImageChange = (e) => {
    const files = Array.from(e.target.files);
    const newImages = files.map((file) => ({
      file,
      preview: URL.createObjectURL(file),
      name: file.name,
      status: 'pending', // 'pending', 'uploaded', 'failed'
    }));
    setUploadedImages((prevImages) => [...prevImages, ...newImages]);
  };
};

```

```

const handleUpload = async (imageToUpload) => {
  // Simulazione di un'API call per l'upload e la classificazione AI
  console.log(`Caricamento immagine: ${imageToUpload.name}`);
  setUploadedImages((prevImages) =>
    prevImages.map((img) =>
      img.name === imageToUpload.name ? { ...img, status: 'uploading' } : img
    )
  );
};

```

```

try {
  // Simula un ritardo di caricamento
  await new Promise((resolve) => setTimeout(resolve, 2000));

  // Simulazione della risposta AI
  const classificationResult = Math.random() > 0.5 ? 'Grado 2 (OA Lieve)' : 'Grado 3 (OA Moderata)';
  const riskCategory = Math.random() > 0.7 ? 'Alto' : 'Basso';

```

```

  setUploadedImages((prevImages) =>
    prevImages.map((img) =>
      img.name === imageToUpload.name
        ? { ...img, status: 'uploaded', classification: classificationResult, risk: riskCategory }
        : img
    )
  );
  alert(`Immagine ${imageToUpload.name} caricata e classificata: ${classificationResult} (Rischio:
${riskCategory})`);
} catch (error) {
  console.error('Errore durante il caricamento:', error);
}

```

```

    setUploadedImages((prevImages) =>
      prevImages.map((img) =>
        img.name === imageToUpload.name ? { ...img, status: 'failed' } : img
      )
    );
    alert(`Errore nel caricamento di ${imageToUpload.name}. Riprova.`);
  }
};

return (
  <div className="p-8 bg-white rounded-lg shadow-xl max-w-3xl mx-auto my-8">
    <h2 className="text-3xl font-bold text-blue-800 mb-6 text-center">Carica Immagini
Radiografiche</h2>
    <div className="border-2 border-dashed border-gray-300 rounded-lg p-6 text-center mb-6
hover:border-blue-500 transition-colors duration-300">
      <input
        type="file"
        id="radiographUpload"
        accept="image/*"
        multiple
        onChange={handleImageChange}
        className="hidden"
      />
      <label
        htmlFor="radiographUpload"
        className="cursor-pointer text-blue-600 hover:text-blue-800 font-semibold text-lg"
      >
        Clicca qui per selezionare le immagini
      </label>
      <p className="text-gray-500 text-sm mt-2">Formati supportati: JPG, PNG, DICOM
(simulato)</p>
    </div>

    {uploadedImages.length > 0 && (
      <div className="mt-8">
        <h3 className="text-2xl font-semibold text-blue-700 mb-4">Immagini Caricate:</h3>
        <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
          {uploadedImages.map((image, index) => (
            <div key={index} className="border border-gray-200 rounded-lg p-4 shadow-sm flex flex-col
items-center">
              <img src={image.preview} alt={image.name} className="max-w-full h-48 object-contain
rounded-md mb-4" />
              <p className="font-medium text-gray-800 text-center break-words">{image.name}</p>
              <p className="text-sm text-gray-600 mt-1">
                Stato: {image.status === 'pending' && 'In attesa'}
                {image.status === 'uploading' && 'Caricamento...'}
                {image.status === 'uploaded' && 'Caricato'}
                {image.status === 'failed' && 'Fallito'}
              </p>
              {image.classification && (
                <p className="text-sm text-blue-600 font-semibold mt-2">Classificazione AI:
{image.classification}</p>
              )}
              {image.risk && (
                <p className="text-sm text-purple-600 font-semibold mt-1">Categoria di Rischio:
{image.risk}</p>
              )}
              {image.status === 'pending' && (
                <button

```

```

        onClick={() => handleUpload(image)}
        className="mt-4 bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4
rounded-full transition-all duration-300"
      >
        Carica e Classifica
      </button>
    ))
    {image.status === 'failed' && (
      <button
        onClick={() => handleUpload(image)}
        className="mt-4 bg-red-500 hover:bg-red-600 text-white font-bold py-2 px-4 rounded-full
transition-all duration-300"
      >
        Riprova Caricamento
      </button>
    )}
  </div>
))}
</div>
</div>
)}
</div>
);
};

```

// Componente Questionario

```

const Questionnaire = ({ questionnaireData, setQuestionnaireData }) => {
  const questions = [
    {
      id: 'painLevel',
      type: 'range',
      label: 'Qual è il tuo livello di dolore al ginocchio (da 0 a 10)?',
      min: 0,
      max: 10,
    },
    {
      id: 'functionality',
      type: 'select',
      label: 'Quanto l\'osteoartrosi influisce sulla tua funzionalità quotidiana?',
      options: [
        { value: '', label: 'Seleziona...' },
        { value: 'none', label: 'Per niente' },
        { value: 'mild', label: 'Lievemente' },
        { value: 'moderate', label: 'Moderatamente' },
        { value: 'severe', label: 'Severamente' },
      ],
    },
    {
      id: 'qualityOfLife',
      type: 'textarea',
      label: 'Come valuti la tua qualità di vita attuale a causa dell\'osteoartrosi?',
    },
    {
      id: 'mobility',
      type: 'radio',
      label: 'Riesci a camminare senza ausili?',
      options: [
        { value: 'yes', label: 'Sì' },
        { value: 'no', label: 'No' },
      ],
    },
  ],

```



```

    ],
  },
];

const handleChange = (e) => {
  const { name, value, type, checked } = e.target;
  setQuestionnaireData((prevData) => ({
    ...prevData,
    [name]: type === 'checkbox' ? checked : value,
  }));
};

const handleSubmit = (e) => {
  e.preventDefault();
  // Qui si invierebbero i dati del questionario al backend
  console.log('Dati questionario salvati:', questionnaireData);
  alert('Questionario compilato con successo!');
};

return (
  <div className="p-8 bg-white rounded-lg shadow-xl max-w-3xl mx-auto my-8">
    <h2 className="text-3xl font-bold text-blue-800 mb-6 text-center">Questionario di
Valutazione</h2>
    <form onSubmit={handleSubmit} className="space-y-6">
      {questions.map((q) => (
        <div key={q.id}>
          <label htmlFor={q.id} className="block text-lg font-medium text-gray-700 mb-2">
            {q.label}
          </label>
          {q.type === 'range' && (
            <>
              <input
                type="range"
                id={q.id}
                name={q.id}
                min={q.min}
                max={q.max}
                value={questionnaireData[q.id] || q.min}
                onChange={handleChange}
                className="w-full h-2 bg-gray-200 rounded-lg appearance-none cursor-pointer range-lg"
              />
              <p className="text-center text-gray-600 mt-2">Valore: {questionnaireData[q.id] || q.min}</p>
            </>
          )}
          {q.type === 'select' && (
            <select
              id={q.id}
              name={q.id}
              value={questionnaireData[q.id] || ""}
              onChange={handleChange}
              className="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500 sm:text-lg"
            >
              {q.options.map((option) => (
                <option key={option.value} value={option.value}>
                  {option.label}
                </option>
              ))}
            </select>
          )}
        </div>
      ))}
    </form>
  </div>
);

```

```

    ))
    {q.type === 'textarea' && (
      <textarea
        id={q.id}
        name={q.id}
        value={questionnaireData[q.id] || ""}
        onChange={handleChange}
        rows="4"
        className="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm
focus:ring-blue-500 focus:border-blue-500 sm:text-lg"
        placeholder="Scrivi qui la tua risposta..."
      ></textarea>
    ))
    {q.type === 'radio' && (
      <div className="mt-2 space-y-2">
        {q.options.map((option) => (
          <div key={option.value} className="flex items-center">
            <input
              type="radio"
              id={` ${q.id}-${option.value}`}
              name={q.id}
              value={option.value}
              checked={questionnaireData[q.id] === option.value}
              onChange={handleChange}
              className="h-5 w-5 text-blue-600 border-gray-300 focus:ring-blue-500"
            />
            <label htmlFor={` ${q.id}-${option.value}`} className="ml-2 text-lg text-gray-700">
              {option.label}
            </label>
          </div>
        ))}
      </div>
    ))
  </div>
  <button
    type="submit"
    className="w-full bg-blue-600 hover:bg-blue-700 text-white font-bold py-3 px-6 rounded-full
shadow-lg transition-all duration-300 transform hover:scale-105"
  >
    Invia Questionario
  </button>
</form>
</div>
);
};

```

// Componente per l'input manuale dei passi (placeholder per pedometro)

```

const PedometerInput = ({ dailySteps, setDailySteps }) => {
  const handleChange = (e) => {
    setDailySteps(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    console.log('Passi giornalieri registrati:', dailySteps);
    alert('Passi giornalieri registrati con successo!');
  };

```

```

return (
  <div className="bg-blue-50 p-6 rounded-xl shadow-md mt-6">
    <h3 className="text-xl font-semibold text-blue-700 mb-4">Registra i tuoi passi giornalieri</h3>
    <form onSubmit={handleSubmit} className="flex flex-col sm:flex-row items-center space-y-4 sm:space-y-0 sm:space-x-4">
      <input
        type="number"
        value={dailySteps || ""}
        onChange={handleChange}
        placeholder="Inserisci i passi"
        className="flex-grow px-4 py-2 border border-gray-300 rounded-md shadow-sm focus:ring-blue-500 focus:border-blue-500 text-lg"
      />
      <button
        type="submit"
        className="bg-blue-600 hover:bg-blue-700 text-white font-bold py-2 px-5 rounded-full shadow-lg transition-all duration-300"
      >
        Salva Passi
      </button>
    </form>
    <p className="text-sm text-gray-600 mt-3">
      (Simulazione: in un'app reale, questo si integrerebbe con il contapassi del tuo smartphone o un dispositivo indossabile.)
    </p>
  </div>
);
};

```

// Componente Piano Terapeutico

```

const TherapyPlan = ({ dailySteps }) => {
  // Esempio di piano terapeutico basato su dati fittizi
  const therapeuticIndications = [
    "Mantieni una dieta equilibrata e ricca di antiossidanti.",
    "Bevi almeno 2 litri d'acqua al giorno.",
    "Evita carichi eccessivi sul ginocchio.",
    "Pratica esercizi a basso impatto regolarmente."
  ];

  const exerciseVideos = [
    {
      id: 1,
      title: 'Esercizio 1: Rinforzo Quadricipiti',
      description: 'Esercizi isometrici per rafforzare i muscoli della coscia.',
      videoUrl: 'https://www.youtube.com/embed/dQw4w9WgXcQ?si=ul31T-5324h01234', // Placeholder video
      instructions: [
        "Siediti con la schiena dritta.",
        "Estendi una gamba in avanti, contraendo il quadricipite.",
        "Mantieni la posizione per 5 secondi, poi rilassa.",
        "Ripeti 10 volte per gamba."
      ]
    },
    {
      id: 2,
      title: 'Esercizio 2: Flessione del Ginocchio',
      description: 'Migliora la flessibilità del ginocchio con movimenti controllati.',
      videoUrl: 'https://www.youtube.com/embed/dQw4w9WgXcQ?si=ul31T-5324h01234', // Placeholder video
    }
  ];
};

```

```

instructions: [
  "Sdraiati sulla schiena con le gambe distese.",
  "Piega lentamente un ginocchio, portando il tallone verso i glutei.",
  "Mantieni la posizione per un attimo, poi estendi lentamente la gamba.",
  "Ripeti 10 volte per gamba."
],
// Aggiungi altri esercizi qui
];

return (
  <div className="p-8 bg-white rounded-lg shadow-xl max-w-4xl mx-auto my-8">
    <h2 className="text-3xl font-bold text-blue-800 mb-6 text-center">Il tuo Piano Terapeutico</h2>

    <div className="mb-8">
      <h3 className="text-2xl font-semibold text-blue-700 mb-4">Indicazioni Igiene e Stile di Vita:</h3>
      <ul className="list-disc list-inside text-lg text-gray-700 space-y-2">
        {therapeuticIndications.map((indication, index) => (
          <li key={index}>{indication}</li>
        ))}
      </ul>
    </div>

    <div className="mb-8">
      <h3 className="text-2xl font-semibold text-blue-700 mb-4">Esercizi Terapeutici con Video:</h3>
      <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
        {exerciseVideos.map((exercise) => (
          <div key={exercise.id} className="border border-gray-200 rounded-lg p-4 shadow-sm">
            <h4 className="text-xl font-medium text-gray-800 mb-3">{exercise.title}</h4>
            <p className="text-gray-600 mb-4">{exercise.description}</p>
            <div className="aspect-w-16 aspect-h-9 mb-4 rounded-md overflow-hidden">
              <iframe
                src={exercise.videoUrl}
                title={exercise.title}
                frameBorder="0"
                allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-
picture"
                allowFullScreen
                className="w-full h-full"
              ></iframe>
            </div>
            <h5 className="text-lg font-semibold text-blue-600 mb-2">Istruzioni:</h5>
            <ol className="list-decimal list-inside text-gray-700 space-y-1">
              {exercise.instructions.map((instruction, idx) => (
                <li key={idx}>{instruction}</li>
              ))}
            </ol>
            <p className="text-sm text-gray-500 mt-4">
              (Nota: Il tracciamento del movimento in tempo reale e il feedback audio/visivo sarebbero
integrati qui in una versione completa.)
            </p>
          </div>
        ))}
      </div>
    </div>

    <div>
      <PedometerInput dailySteps={dailySteps} setDailySteps={() => {}} /> {/* Passa una funzione vuota
per setDailySteps qui, dato che PedometerInput gestisce il proprio stato interno */}
    </div>
  </div>

```

```

);
};

// Componente principale dell'applicazione
const App = () => {
  const [currentPage, setCurrentPage] = useState('home');
  const [userData, setUserData] = useState({});
  const [consentGiven, setConsentGiven] = useState(false);
  const [uploadedImages, setUploadedImages] = useState([]);
  const [questionnaireData, setQuestionnaireData] = useState({});
  const [dailySteps, setDailySteps] = useState(""); // Stato per i passi giornalieri

  // Funzione per navigare tra le pagine
  const handleNavigate = (page) => {
    setCurrentPage(page);
  };

  // Renderizza il contenuto della pagina corrente
  const renderPage = () => {
    switch (currentPage) {
      case 'home':
        return <Home />;
      case 'profile':
        return <Profile userData={userData} setUserData={setUserData} />;
      case 'consent':
        return <ConsentForm consentGiven={consentGiven} setConsentGiven={setConsentGiven} />;
      case 'imageUpload':
        return <ImageUpload uploadedImages={uploadedImages}
setUploadedImages={setUploadedImages} />;
      case 'questionnaire':
        return <Questionnaire questionnaireData={questionnaireData}
setQuestionnaireData={setQuestionnaireData} />;
      case 'therapyPlan':
        return <TherapyPlan dailySteps={dailySteps} setDailySteps={setDailySteps} />;
      default:
        return <Home />;
    }
  };

  return (
    <div className="min-h-screen flex flex-col font-inter bg-gray-100">
      { /* Script per Tailwind CSS */ }
      <script src="https://cdn.tailwindcss.com"></script>
      { /* Configurazione Tailwind per il font Inter */ }
      <style>
        {
          @import
url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap');
          body {
            font-family: 'Inter', sans-serif;
          }
          /* Per la corretta visualizzazione degli iframe (video) */
          .aspect-w-16 {
            position: relative;
            width: 100%;
          }
          .aspect-h-9 {
            padding-bottom: 56.25%; /* 9 / 16 * 100 */
          }
        }
      </style>
    </div>
  );
};

```

```
.aspect-w-16 iframe {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
}  
`}  
</style>
```

```
<Header currentPage={currentPage} onNavigate={handleNavigate} />  
<main className="flex-grow container mx-auto p-4">  
  {renderPage()}  
</main>  
<Footer />  
</div>
```

```
);  
};
```

```
export default App;
```