Q1. import React, { useState } from 'react';

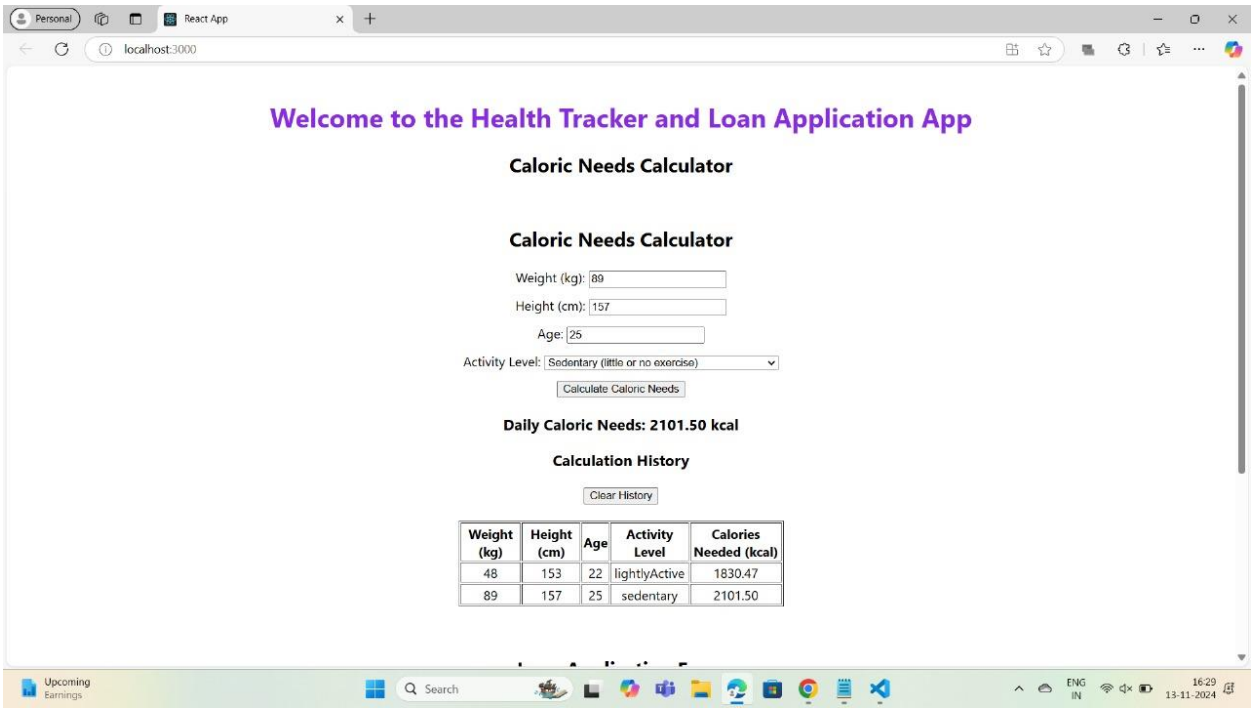
```
const CaloricNeedsCalculator = () => {
  const [weight, setWeight] = useState('');
  const [height, setHeight] = useState('');
  const [age, setAge] = useState('');
  const [activityLevel, setActivityLevel] = useState('');
  const [calories, setCalories] = useState(null);
  const [history, setHistory] = useState([]);

  const calculateCalories = () => {
    if (weight && height && age && activityLevel) {
      // Calculate Basal Metabolic Rate (BMR) using Mifflin-St Jeor Equation
      const bmr = 10 * weight + 6.25 * height - 5 * age + 5; // For males; for females, subtract 161 instead of adding 5.

      // Multiply BMR by activity factor
      const activityFactors = {
        sedentary: 1.2,
        lightlyActive: 1.375,
        moderatelyActive: 1.55,
        veryActive: 1.725,
        superActive: 1.9,
      };

      const caloricNeeds = (bmr * activityFactors[activityLevel]).toFixed(2);
      setCalories(caloricNeeds);

      // Add the calculation to history
```

```
    setHistory([
      ...history,
      { weight, height, age, activityLevel, calories: caloricNeeds },
    ]);
  } else {
    alert('Please enter all fields');
  }
};

const clearHistory = () => {
  setHistory([]);
};

return (
  <div style={{ padding: '20px', maxWidth: '400px', margin: 'auto', textAlign: 'center' }}>
    <h2>Caloric Needs Calculator</h2>

    <div style={{ marginBottom: '10px' }}>
      <label>Weight (kg): </label>
      <input
        type="number"
        value={weight}
        onChange={(e) => setWeight(e.target.value)}
      />
    </div>

    <div style={{ marginBottom: '10px' }}>
      <label>Height (cm): </label>
      <input
```

```
        type="number"
        value={height}
        onChange={(e) => setHeight(e.target.value)}
      />
    </div>

    <div style={{ marginBottom: '10px' }}>
      <label>Age: </label>
      <input
        type="number"
        value={age}
        onChange={(e) => setAge(e.target.value)}
      />
    </div>

    <div style={{ marginBottom: '10px' }}>
      <label>Activity Level: </label>
      <select value={activityLevel} onChange={(e) => setActivityLevel(e.target.value)}>
        <option value="">Select</option>
        <option value="sedentary">Sedentary (little or no exercise)</option>
        <option value="lightlyActive">Lightly active (light exercise/sports 1-3 days/week)</option>
        <option value="moderatelyActive">Moderately active (moderate exercise/sports 3-5
days/week)</option>
        <option value="veryActive">Very active (hard exercise/sports 6-7 days a week)</option>
        <option value="superActive">Super active (very hard exercise & a physical job)</option>
      </select>
    </div>

    <button onClick={calculateCalories}>Calculate Caloric Needs</button>
```

```
{calories && (
  <div style={{ marginTop: '20px' }}>
    <h3>Daily Caloric Needs: {calories} kcal</h3>
  </div>
)}

<h3>Calculation History</h3>
<button onClick={clearHistory} style={{ marginBottom: '10px' }}>Clear History</button>
<table border="1" style={{ width: '100%', marginTop: '10px' }}>
  <thead>
    <tr>
      <th>Weight (kg)</th>
      <th>Height (cm)</th>
      <th>Age</th>
      <th>Activity Level</th>
      <th>Calories Needed (kcal)</th>
    </tr>
  </thead>
  <tbody>
    {history.map((entry, index) => (
      <tr key={index}>
        <td>{entry.weight}</td>
        <td>{entry.height}</td>
        <td>{entry.age}</td>
        <td>{entry.activityLevel}</td>
        <td>{entry.calories}</td>
      </tr>
    ))}
```

```
      </tbody>
    </table>
  </div>
 );
};

export default CaloricNeedsCalculator;
```



Q2. import React, { useState } from 'react';

```
const LoanApplicationForm = () => {
  const [step, setStep] = useState(1); // Keeps track of the current step
  const [personalInfo, setPersonalInfo] = useState({ name: '', email: '', phone: '' });
  const [loanDetails, setLoanDetails] = useState({ amount: '', term: '', purpose: '' });
  const [incomeVerification, setIncomeVerification] = useState({ income: '', document: '' });
  const [errors, setErrors] = useState({});
```

```
// Validate Personal Information
const validatePersonalInfo = () => {
  const errors = {};
  if (!personalInfo.name) errors.name = 'Name is required';
  if (!personalInfo.email) errors.email = 'Email is required';
  if (!personalInfo.phone) errors.phone = 'Phone number is required';
  setErrors(errors);
  return Object.keys(errors).length === 0;
};

// Validate Loan Details
const validateLoanDetails = () => {
  const errors = {};
  if (!loanDetails.amount) errors.amount = 'Loan amount is required';
  if (!loanDetails.term) errors.term = 'Loan term is required';
  if (!loanDetails.purpose) errors.purpose = 'Purpose of loan is required';
  setErrors(errors);
  return Object.keys(errors).length === 0;
};

// Validate Income Verification
const validateIncomeVerification = () => {
  const errors = {};
  if (!incomeVerification.income) errors.income = 'Income is required';
  if (!incomeVerification.document) errors.document = 'Income verification document is required';
  setErrors(errors);
  return Object.keys(errors).length === 0;
};
```

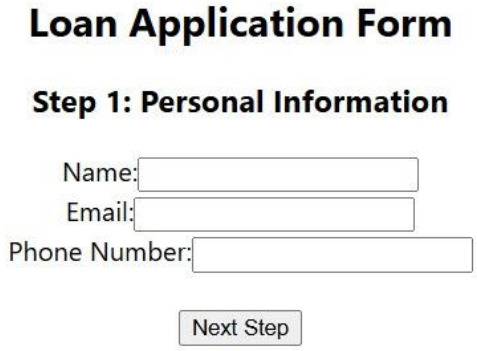
```
// Handle Next Step
const handleNext = () => {
  if (step === 1 && validatePersonalInfo()) {
    setStep(step + 1);
  } else if (step === 2 && validateLoanDetails()) {
    setStep(step + 1);
  } else if (step === 3 && validateIncomeVerification()) {
    alert('Loan application submitted successfully!');
  }
};

// Handle Input Change
const handleChange = (e, step) => {
  const { name, value } = e.target;

  if (step === 1) {
    setPersonalInfo({ ...personalInfo, [name]: value });
  } else if (step === 2) {
    setLoanDetails({ ...loanDetails, [name]: value });
  } else if (step === 3) {
    setIncomeVerification({ ...incomeVerification, [name]: value });
  }
};

return (
  <div style={{ maxWidth: '600px', margin: 'auto', padding: '20px' }}>
    <h2>Loan Application Form</h2>
    <form>
      {step === 1 && (
```

```
<div>
 <h3>Step 1: Personal Information</h3>
 <div>
  <label>Name:</label>
  <input
   type="text"
   name="name"
   value={personalInfo.name}
   onChange={(e) => handleChange(e, 1)}
  />
  {errors.name && <p style={{ color: 'red' }}>{errors.name}</p>}
 </div>
 <div>
  <label>Email:</label>
  <input
   type="email"
   name="email"
   value={personalInfo.email}
   onChange={(e) => handleChange(e, 1)}
  />
  {errors.email && <p style={{ color: 'red' }}>{errors.email}</p>}
 </div>
 <div>
  <label>Phone Number:</label>
  <input
   type="tel"
   name="phone"
   value={personalInfo.phone}
   onChange={(e) => handleChange(e, 1)}
```

```
    />
    {errors.phone && <p style={{ color: 'red' }}>{errors.phone}</p>}
   </div>
  </div>
)}

{step === 2 && (
  <div>
   <h3>Step 2: Loan Details</h3>
   <div>
    <label>Loan Amount:</label>
    <input
     type="number"
     name="amount"
     value={loanDetails.amount}
     onChange={(e) => handleChange(e, 2)}
    />
    {errors.amount && <p style={{ color: 'red' }}>{errors.amount}</p>}
   </div>
   <div>
    <label>Loan Term (Years):</label>
    <input
     type="number"
     name="term"
     value={loanDetails.term}
     onChange={(e) => handleChange(e, 2)}
    />
    {errors.term && <p style={{ color: 'red' }}>{errors.term}</p>}
   </div>
```

```
   <div>
    <label>Purpose of Loan:</label>
    <input
     type="text"
     name="purpose"
     value={loanDetails.purpose}
     onChange={(e) => handleChange(e, 2)}
    />
    {errors.purpose && <p style={{ color: 'red' }}>{errors.purpose}</p>}
   </div>
  </div>
)}

{step === 3 && (
  <div>
   <h3>Step 3: Income Verification</h3>
   <div>
    <label>Monthly Income:</label>
    <input
     type="number"
     name="income"
     value={incomeVerification.income}
     onChange={(e) => handleChange(e, 3)}
    />
    {errors.income && <p style={{ color: 'red' }}>{errors.income}</p>}
   </div>
   <div>
    <label>Income Verification Document:</label>
    <input
```

```
            type="file"
            name="document"
            onChange={(e) => handleChange(e, 3)}
           />
           {errors.document && <p style={{ color: 'red' }}>{errors.document}</p>}
          </div>
         </div>
        )}

        <div style={{ marginTop: '20px' }}>
         <button type="button" onClick={handleNext}>
          {step === 3 ? 'Submit Application' : 'Next Step'}
         </button>
        </div>
       </form>
      </div>
    );
};

export default LoanApplicationForm;
```



Loan Application Form

Step 2: Loan Details

Loan Amount: 1000000

Loan Term (Years): 5

Purpose of Loan: NA

Next Step

Loan Application Form

Step 3: Income Verification

Monthly Income: 850000

Income Verification Document: Choose File foolproof.webp

Submit Application