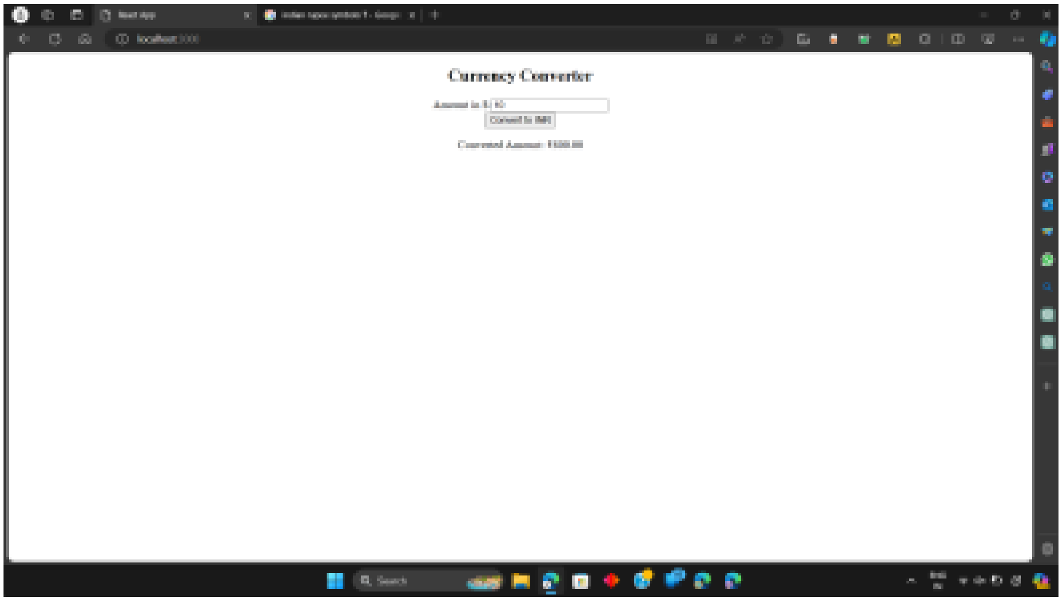
1.

```
import React, { useState } from 'react';
import './App.css';

const CurrencyConverter = () => {

  const [amount, setAmount] = useState('');
const [convertedAmount, setConvertedAmount] =
useState('');


  const handleAmountChange = (event) => {
    const inputAmount = event.target.value;
    setAmount(inputAmount);
  };


  const convertCurrency = () => {
    const exchangeRate = 80;
    const convertedAmt = amount * exchangeRate;
  setConvertedAmount(convertedAmt.toFixed(2));
  };

  return (
    <div>
      <h2>Currency Converter</h2>
      <label>
        Amount in $:
        <input type="number" value={amount}
onChange={handleAmountChange} />
```

```
      </label>
      <br />
      <button onClick={convertCurrency}>Convert to
INR</button>
      <br />
      <p>Converted Amount: ₹{convertedAmount} </p>
    </div>
  );
};

function App() {
  return (
    <div className="App">
      <CurrencyConverter />
    </div>
  );
}

export default App;
```



Currency Converter

Amount in $: 10

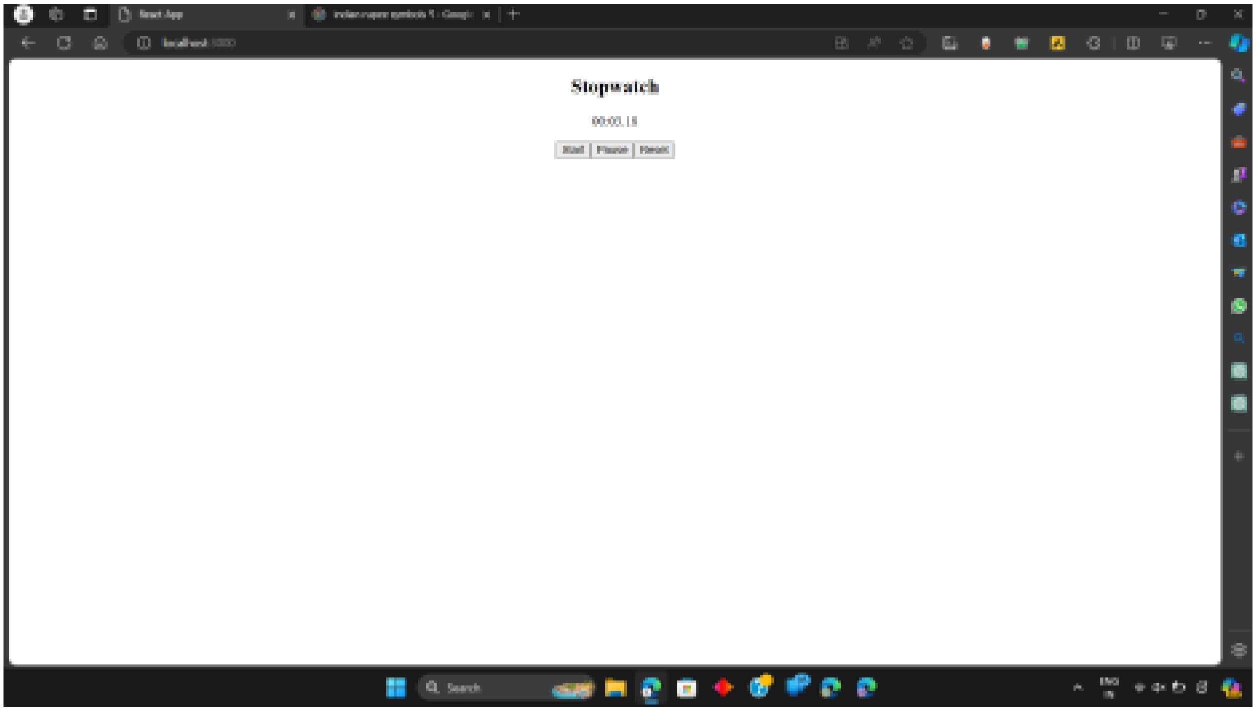
Convert to INR

Converted Amount: ₹830.00

2.

```
import React, { useState, useRef } from 'react';
import './App.css';

const Stopwatch = () => {
  const [time, setTime] = useState(0);
  const [isRunning, setIsRunning] = useState(false);
  const intervalRef = useRef(null);

  const startTimer = () => {
    if (!isRunning) {
      const startTime = Date.now() - time;
      intervalRef.current = setInterval(() => {
        setTime(Date.now() - startTime);
      }, 10);
      setIsRunning(true);
    }
  };

  const pauseTimer = () => {
    if (isRunning) {
      clearInterval(intervalRef.current);
      setIsRunning(false);
    }
  };

  const resetTimer = () => {
    clearInterval(intervalRef.current);
    setTime(0);
    setIsRunning(false);
  };
  const formatTime = (milliseconds) => {
    const totalSeconds = Math.floor(milliseconds / 1000);
    const minutes = Math.floor(totalSeconds / 60); const
    seconds = totalSeconds % 60;
    const ms = Math.floor((milliseconds % 1000) / 10);
    return `${minutes < 10 ? '0' :
''}${minutes}:${seconds < 10 ? '0' : ''}${seconds}.${ms
< 10 ? '0' : ''}${ms}`;
  };

  return (
    <div>
      <h2>Stopwatch</h2>
      <p>{formatTime(time)}</p>
      <button onClick={startTimer}>Start</button>
    <button onClick={pauseTimer}>Pause</button>
```

```
      <button onClick={resetTimer}>Reset</button>
      </div>
    );
  };

function App() {
  return (
    <div className="App">
      <Stopwatch />
    </div>
  );
}

export default App;
```

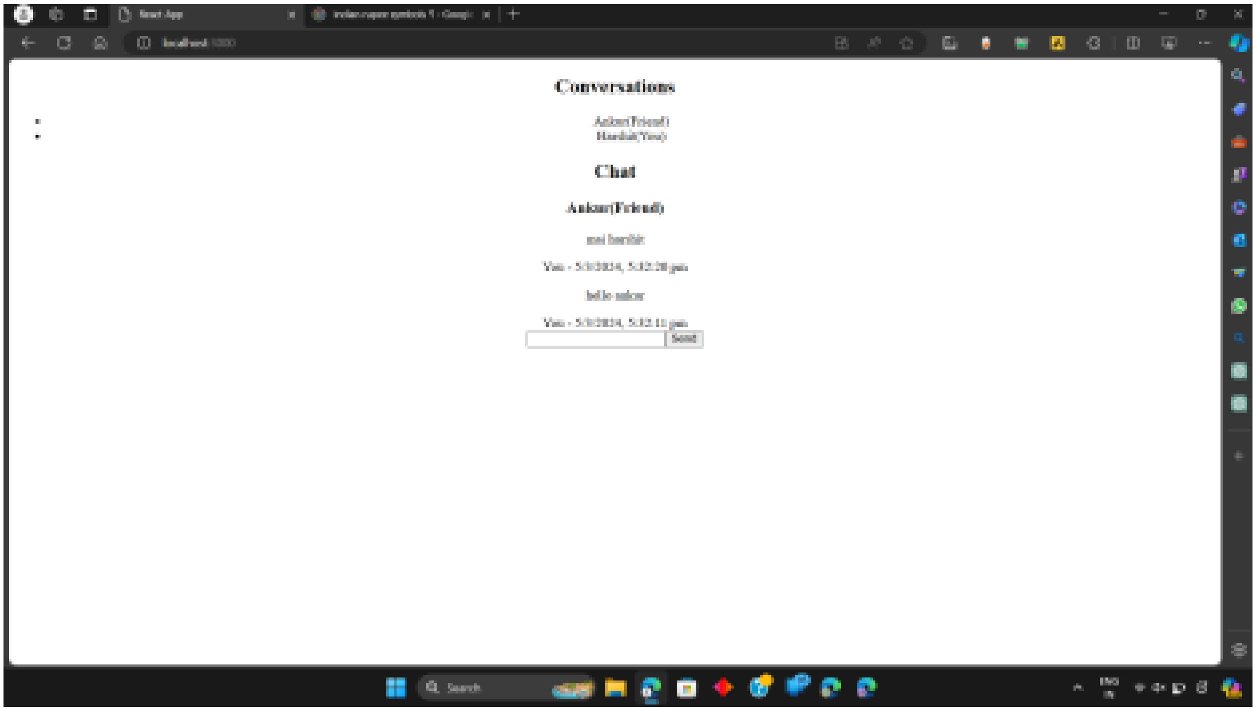


Stopwatch

00:03.18

3.

```
import React, { useState, useEffect } from 'react';
import './App.css';


function App() {
  const [conversations, setConversations] = useState([]);
  const [selectedConversation, setSelectedConversation] = useState(null);
  const [newMessage, setNewMessage] = useState('');
  const [messages, setMessages] = useState([]);


  useEffect(() => {
    fetchConversations();
  }, []);


  useEffect(() => {
    if (selectedConversation) {
      fetchMessages(selectedConversation.id);
    }
  }, [selectedConversation]);


  const fetchConversations = () => {
    const mockConversations = [
      { id: 1, name: 'Ankur(Friend)' },
      { id: 2, name: 'Harshit(You)' },
    ];
    setConversations(mockConversations);
  };
  const fetchMessages = (conversationId) => {
    const mockMessages = [

    ];
    setMessages(mockMessages);
  };


  const handleConversationClick = (conversation) => {
    setSelectedConversation(conversation);
  };


  const handleMessageSend = () => {
    const message = { id: messages.length + 1, text: newMessage, sender: 'You', timestamp: new Date()
};
    setMessages([message, ...messages]);
    setNewMessage('');
  };


  return (
    <div className="App">
      <div className="sidebar">
        <h2>Conversations</h2>
        <ul>
          {conversations.map((conversation) => (
            <li key={conversation.id} onClick={() => handleConversationClick(conversation)}>
              {conversation.name}
            </li>
          ))}
        </ul>
      </div>
      <div className="chat">
        <h2>Chat</h2>
        {selectedConversation && (
          <div>
            <h3>{selectedConversation.name}</h3>
            <div className="messages">
              {messages.map((message) => (
                <div key={message.id} className={message.sender === 'You' ? 'sent' : 'received'}>
                  <p>{message.text}</p>
                  <span>{message.sender} - {message.timestamp.toLocaleString()}</span>
                </div>
              ))}
            </div>
            <div className="message-input">
              <input type="text" value={newMessage} onChange={(e) => setNewMessage(e.target.value)} />
              <button onClick={handleMessageSend}>Send</button>
            </div>
          </div>
        )}
      </div>
    </div>
  );
}
export default App;
```



Conversations

Ankur(Friend)
Harshit(You)

Chat

Ankur(Friend)

mai hardik

You - 5/3/2024, 5:32:38 pm

hello ankur

You - 5/3/2024, 5:32:11 pm