Name – Ranjan Kumar Pandit

Roll No. – 22it3037

Lab – 8

Using ReactJs

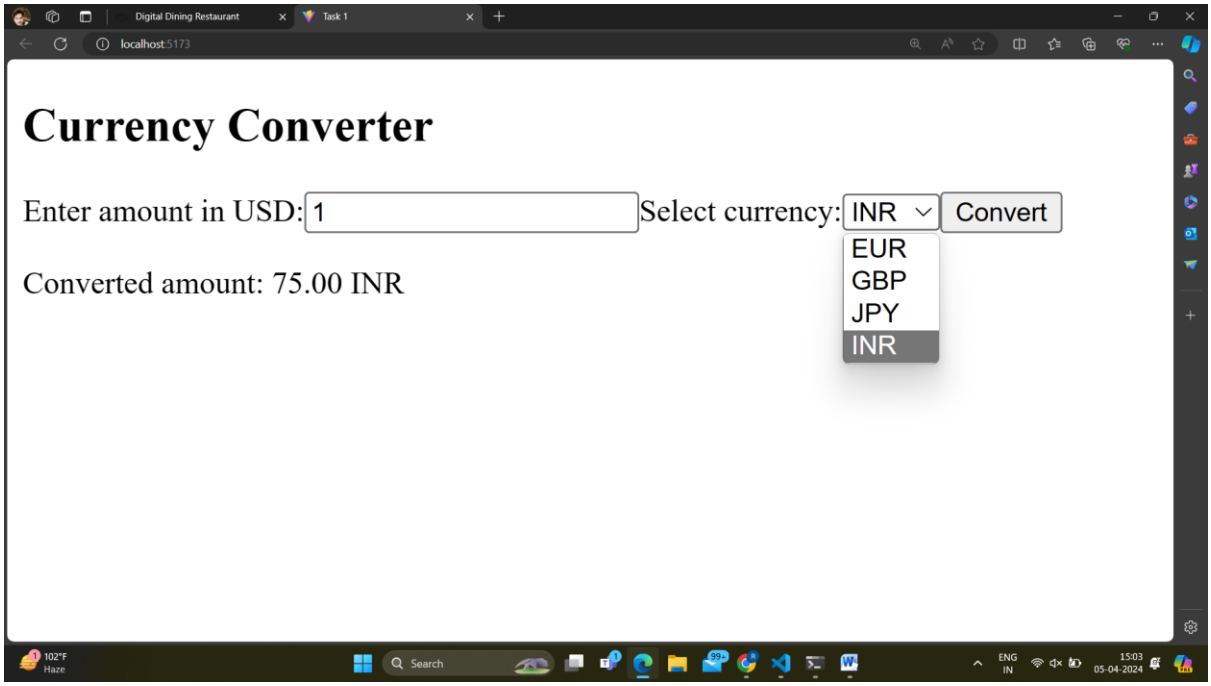
Q1.

CurrencyConvertor.jsx:

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

function CurrencyConverter() {
  const [usdAmount, setUSDAmount] = useState('');
  const [selectedCurrency, setSelectedCurrency] = useState('EUR');
  const [convertedAmount, setConvertedAmount] = useState('');

  const exchangeRates = {
    EUR: 0.85,
  };

  const handleUSDChange = (event) => {
    setUSDAmount(event.target.value);
  };

  const handleCurrencyChange = (event) => {
    setSelectedCurrency(event.target.value);
  };

  const convertCurrency = () => {
    const rate = exchangeRates[selectedCurrency];
    const convertedAmount = parseFloat(usdAmount) * rate;
    setConvertedAmount(convertedAmount.toFixed(2));
  };

  return (
    <div>
      <h2>Currency Converter</h2>
      <div>
        <label>
          Enter amount in USD:
          <input type="number" value={usdAmount} onChange={handleUSDChange} />
        </label>
        <label>
          Select currency:
```

```
          <select value={selectedCurrency} onChange={handleCurrencyChange}>
            {Object.keys(exchangeRates).map(currency => (
              <option key={currency} value={currency}>{currency}</option>
            ))}
          </select>
        </label>
        <button onClick={convertCurrency}>Convert</button>
      </div>
      {convertedAmount && (
        <div>
          <p>Converted amount: {convertedAmount} {selectedCurrency}</p>
        </div>
      )}
    </div>
  );
}

export default CurrencyConverter;
```

Output:-



Q2.

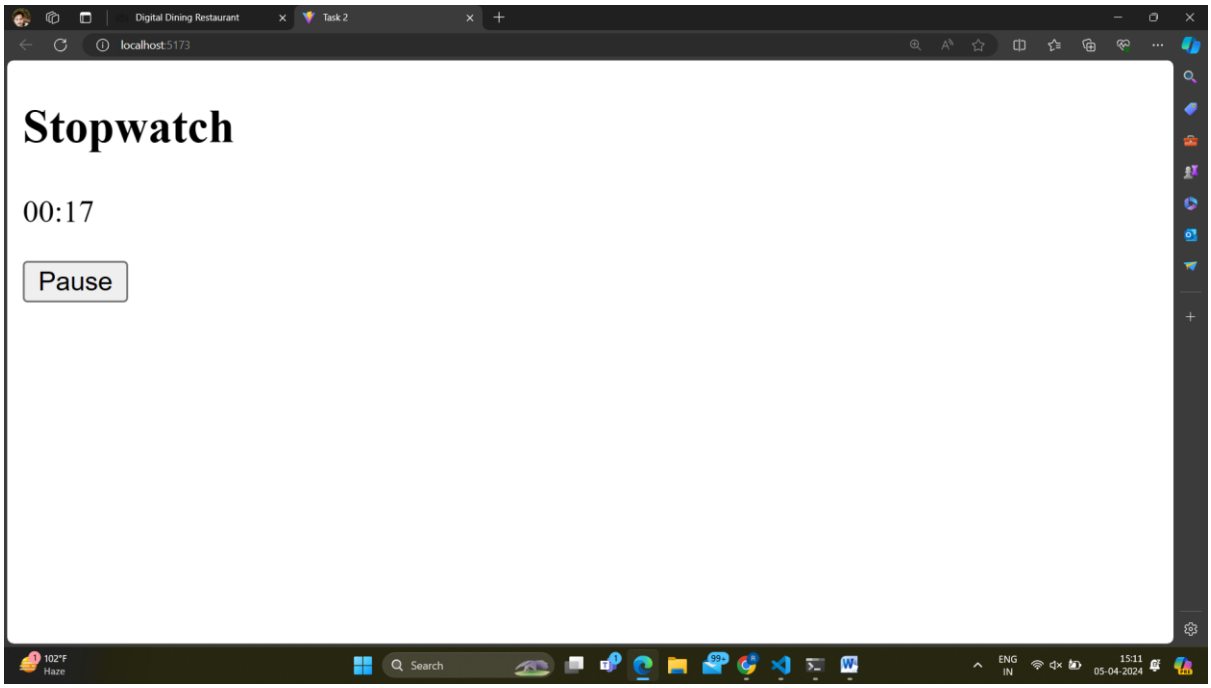
Stopwatch.jsx

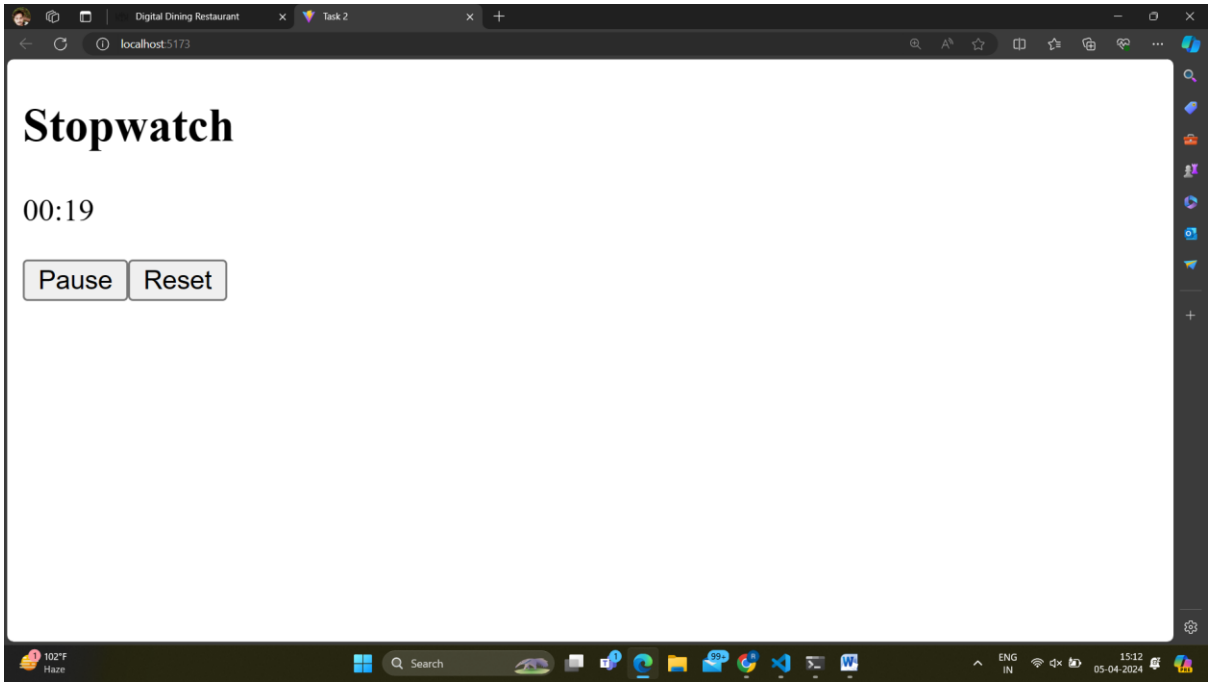
```
import React, { useState, useRef } from 'react';
```

```jsx
function Stopwatch() {
  const [timer, setTimer] = useState(0);
  const [isActive, setIsActive] = useState(false);
  const [isPaused, setIsPaused] = useState(false);
  const intervalRef = useRef(null);

  const handleStart = () => {
    setIsActive(true);
    setIsPaused(false);
    intervalRef.current = setInterval(() => {
      setTimer((prevTimer) => prevTimer + 1);
    }, 1000);
  };

  const handlePause = () => {
    clearInterval(intervalRef.current);
    setIsPaused(true);
  };

  const handleReset = () => {
    clearInterval(intervalRef.current);
    setIsActive(false);
    setIsPaused(false);
    setTimer(0);
  };

  const formatTime = (time) => {
    const minutes = Math.floor(time / 60).toString().padStart(2, '0');
    const seconds = (time % 60).toString().padStart(2, '0');
    return `${minutes}:${seconds}`;
  };

  return (
    <div>
      <h2>Stopwatch</h2>
      <p>{formatTime(timer)}</p>
      <div>
        {!isActive && !isPaused && (
          <button onClick={handleStart}>Start</button>
        )}
        {isActive && (
          <button onClick={handlePause}>Pause</button>
        )}
        {(!isActive || isPaused) && timer !== 0 && (
          <button onClick={handleReset}>Reset</button>
        )}
      </div>
    </div>
```

```
  );
}

export default Stopwatch;
```

Output:-





Using VueJs

Q1.

CurrencyConverter.vue:

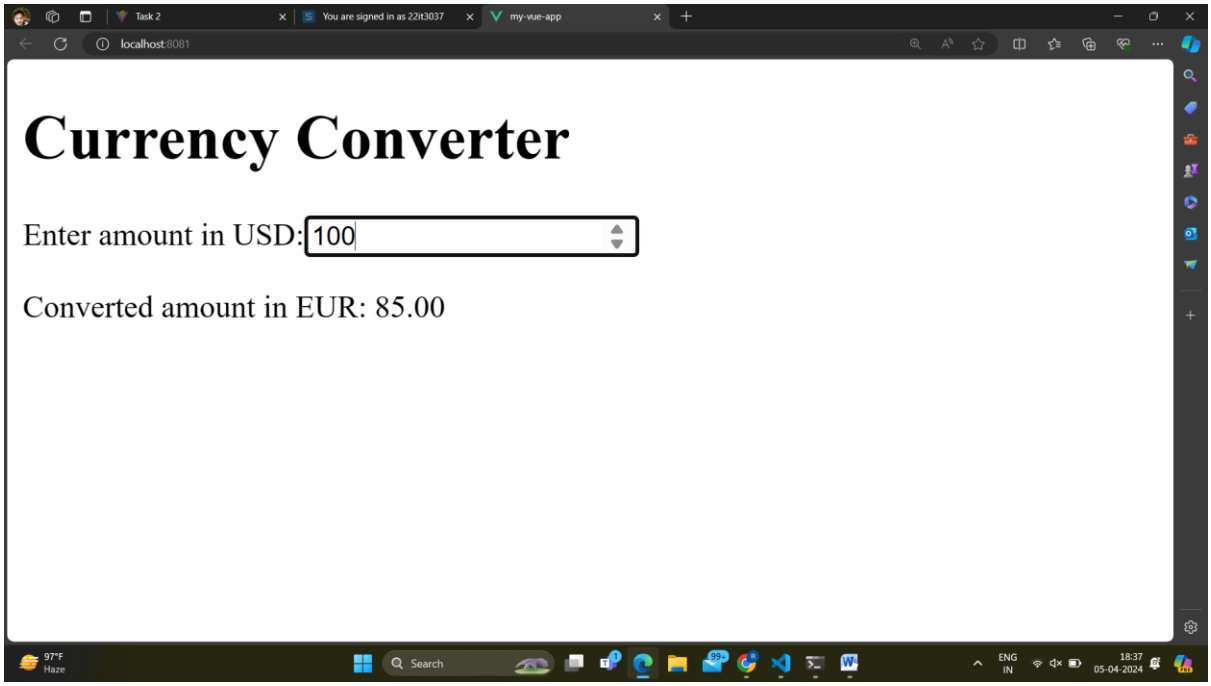
```
<!-- CurrencyConverter.vue -->
<template>
  <div>
    <h1>Currency Converter</h1>
    <label for="usdAmount">Enter amount in USD:</label>
    <input type="number" id="usdAmount" v-model="usdAmount"
@input="convertCurrency">
    <p>Converted amount in EUR: {{ eurAmount }}</p>
  </div>
</template>

<script>
export default {
  data() {
    return {
      usdAmount: '',
      eurAmount: ''
    };
  },
  methods: {
    convertCurrency() {
      const exchangeRate = 0.85;
      this.eurAmount = (parseFloat(this.usdAmount) * exchangeRate).toFixed(2);
    }
  }
};
</script>
```

App.vue:

```
<!-- App.vue -->
<template>
  <div id="app">
    <CurrencyConverter />
  </div>
</template>

<script>
import CurrencyConverter from './components/CurrencyConverter.vue';
```

```
export default {
  name: 'App',
  components: {
    CurrencyConverter
  }
};
</script>
```

Output:-



Q2.

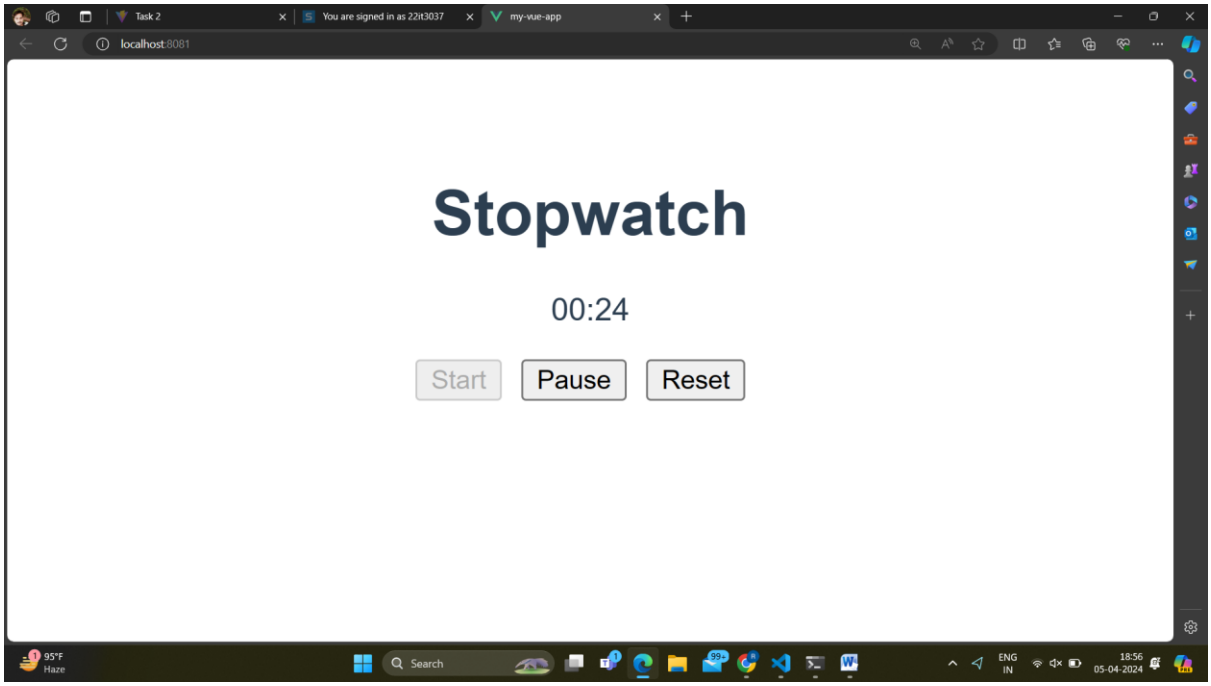
StopwatchTimer.vue:

```
<template>
  <div>
    <h1>Stopwatch</h1>
    <p>{{ formatTime }}</p>
    <div>
      <button @click="startTimer" :disabled="isRunning">Start</button>
      <button @click="pauseTimer" :disabled="!isRunning">Pause</button>
      <button @click="resetTimer">Reset</button>
    </div>
  </div>
</template>
```

```
<script>
export default {
  data() {
    return {
      startTime: null,
      currentTime: 0,
      timerId: null,
      isRunning: false
    };
  },
  computed: {
    formatTime() {
      const minutes = Math.floor(this.currentTime / 60).toString().padStart(2,
'0');
      const seconds = (this.currentTime % 60).toString().padStart(2, '0');
      return `${minutes}:${seconds}`;
    }
  },
  methods: {
    startTimer() {
      if (!this.isRunning) {
        this.startTime = Date.now() - this.currentTime * 1000;
        this.timerId = setInterval(() => {
          this.currentTime = Math.floor((Date.now() - this.startTime) / 1000);
        }, 1000);
        this.isRunning = true;
      }
    },
    pauseTimer() {
      clearInterval(this.timerId);
      this.isRunning = false;
    },
    resetTimer() {
      clearInterval(this.timerId);
      this.isRunning = false;
      this.currentTime = 0;
    }
  }
};
</script>

<style scoped>
button {
  margin-right: 10px;
}
</style>
```

App.vue:

```
<template>
  <div id="app">
    <StopwatchTimer />
  </div>
</template>

<script>
import StopwatchTimer from './components/StopwatchTimer.vue';

export default {
  name: 'App',
  components: {
    StopwatchTimer
  }
};
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```


Task 2
You are signed in as 22it3037
my-vue-app
localhost:8081
Stopwatch
00:24
Start
Pause
Reset
95°F
Haze
Search
ENG
IN
18:56
05-04-2024

Thanks