

Name – Sayan Mukhopadhyay  
Roll No. – 22CS2021

Q1

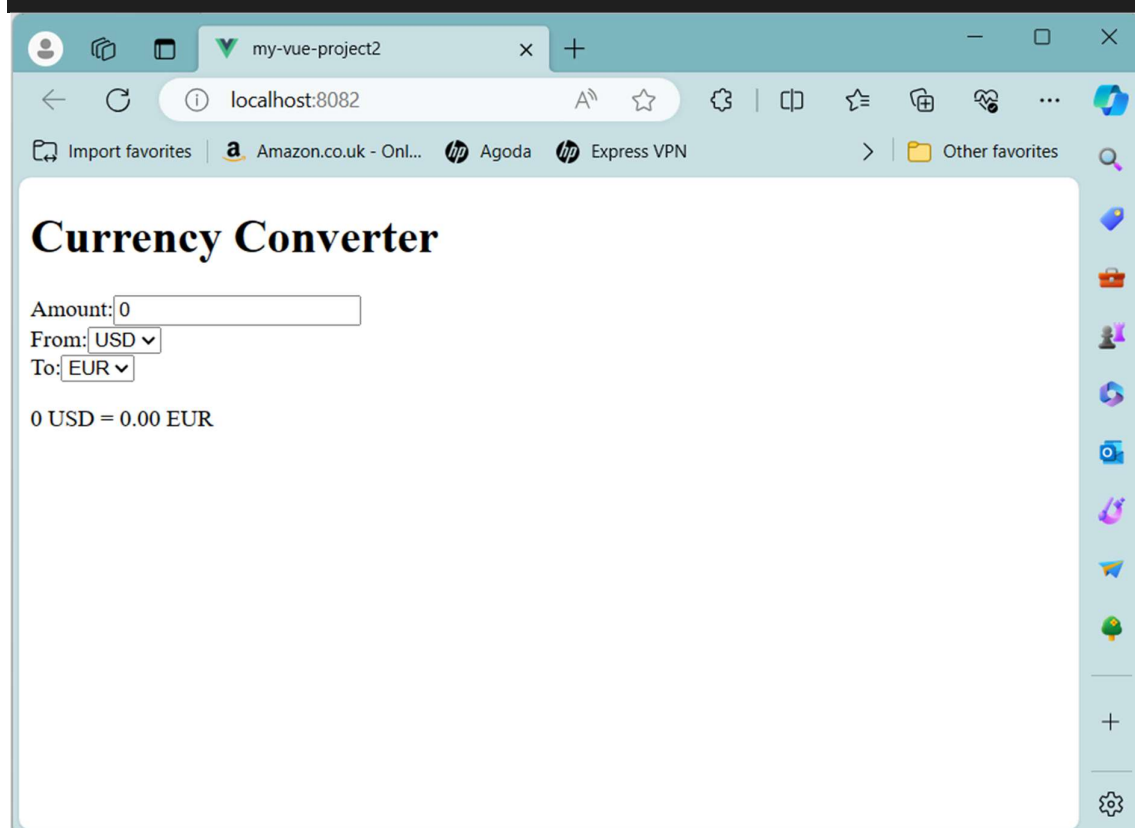
```
<template>
  <div>
    <h1>Currency Converter</h1>
    <div>
      <label for="amount">Amount:</label>
      <input type="number" id="amount" v-model="amount"
@input="convert">
    </div>
    <div>
      <label for="from">From:</label>
      <select id="from" v-model="from" @change="convert">
        <option v-for="(rate, currency) in exchangeRates"
:value="currency" :key="currency">{{ currency }}</option>
      </select>
    </div>
    <div>
      <label for="to">To:</label>
      <select id="to" v-model="to" @change="convert">
        <option v-for="(rate, currency) in exchangeRates"
:value="currency" :key="currency">{{ currency }}</option>
      </select>
    </div>
    <div>
      <p>{{ amount }} {{ from }} = {{ convertedAmount }} {{ to }}</p>
    </div>
  </div>
</template>

<script>
export default {
  data() {
    return {
      amount: 0,
      from: 'USD',
      to: 'EUR',
      exchangeRates: {
        USD: 1,
        EUR: 0.85,
        GBP: 0.72,
        JPY: 112.08,
        AUD: 1.33
      }
    }
  }
}
```

```

        // Add more currencies and their exchange rates here
      },
      // Remove duplicate key
    }
  },
  computed: {
    convertedAmount: function() {
      return (this.amount * this.exchangeRates[this.to] /
this.exchangeRates[this.from]).toFixed(2);
    }
  },
  methods: {
    convert: function() {
      this.convertedAmount = (this.amount * this.exchangeRates[this.to]
/ this.exchangeRates[this.from]).toFixed(2);
    }
  }
}
</script>

```



Q2

```

<template>
  <div class="container">

```

```

    <h1>{{ formatTime }}</h1>
    <button class="button" @click="startPause">{{ isRunning ? 'Pause' :
'Start' }}</button>
    <button class="button" @click="reset">Reset</button>
  </div>
</template>

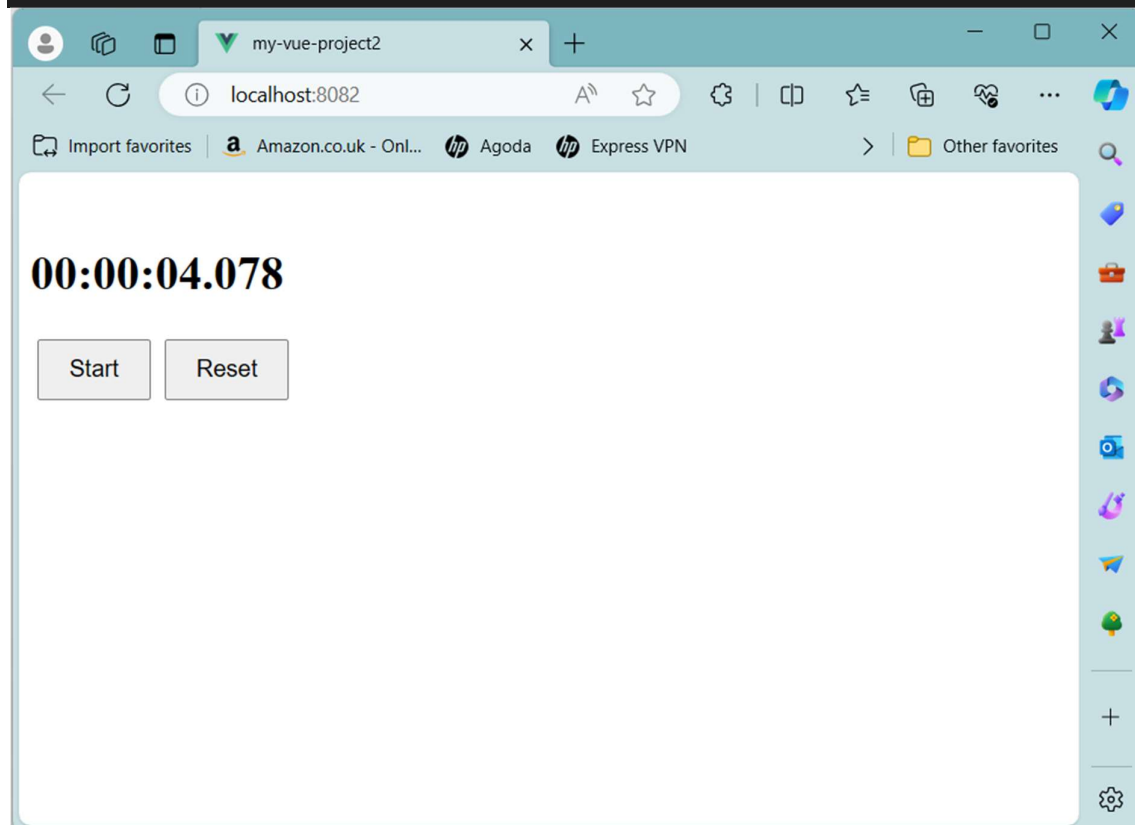
<script>
export default {
  data() {
    return {
      isRunning: false,
      startTime: null,
      currentTime: null
    };
  },
  computed: {
    elapsedTime() {
      if (!this.startTime) return 0;
      return this.currentTime - this.startTime;
    },
    formatTime() {
      const milliseconds = this.elapsedTime % 1000;
      const seconds = Math.floor((this.elapsedTime / 1000) % 60);
      const minutes = Math.floor((this.elapsedTime / (1000 * 60)) %
60);
      const hours = Math.floor((this.elapsedTime / (1000 * 60 * 60)) %
24);

      return `${hours.toString().padStart(2,
'0')}:${minutes.toString().padStart(2,
'0')}:${seconds.toString().padStart(2,
'0')}:${milliseconds.toString().padStart(3, '0')}`;
    }
  },
  methods: {
    startPause() {
      if (this.isRunning) {
        clearInterval(this.timer);
      } else {
        this.startTime = new Date() - this.elapsedTime;
        this.timer = setInterval(() => {
          this.currentTime = new Date();
        }, 10);
      }
      this.isRunning = !this.isRunning;
    }
  }
};

```

```
    },
    reset() {
      this.isRunning = false;
      clearInterval(this.timer);
      this.startTime = null;
      this.currentTime = null;
    }
  }
};
</script>

<style scoped>
.container {
  margin-top: 50px;
}
.button {
  margin: 5px;
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
}
</style>
```



Q3.

```
<template>
  <div class="messaging-app">
    <div class="conversation-list">
      <h2>Conversations</h2>
      <ul>
        <li v-for="conversation in conversations" :key="conversation.id" @click="selectConversation(conversation)">
          {{ conversation.name }}
        </li>
      </ul>
    </div>
    <div class="chat">
      <h2 v-if="selectedConversation">Chat with {{ selectedConversation.name }}</h2>
      <div class="message-list" v-if="selectedConversation">
        <div v-for="message in selectedConversation.messages" :key="message.id" class="message">
          <p>{{ message.text }}</p>
          <small>{{ message.sender }}</small>
        </div>
      </div>
      <div class="message-input" v-if="selectedConversation">
        <textarea v-model="newMessage" placeholder="Type your message here"></textarea>
        <button @click="sendMessage">Send</button>
      </div>
    </div>
  </div>
</template>
```

```
<script>
export default {
  data() {
    return {
      conversations: [
        { id: 1, name: 'Alice', messages: [] },
        { id: 2, name: 'Bob', messages: [] },
        { id: 3, name: 'Charlie', messages: [] }
      ],
      selectedConversation: null,
      newMessage: ''
    };
  },
  methods: {
    selectConversation(conversation) {
      this.selectedConversation = conversation;
    },
    sendMessage() {
      const message = {
        text: this.newMessage,
        sender: 'User' // Assuming the user's name is 'User'
      };
      this.selectedConversation.messages.push(message);
      this.newMessage = '';
    }
  }
};
</script>
```

