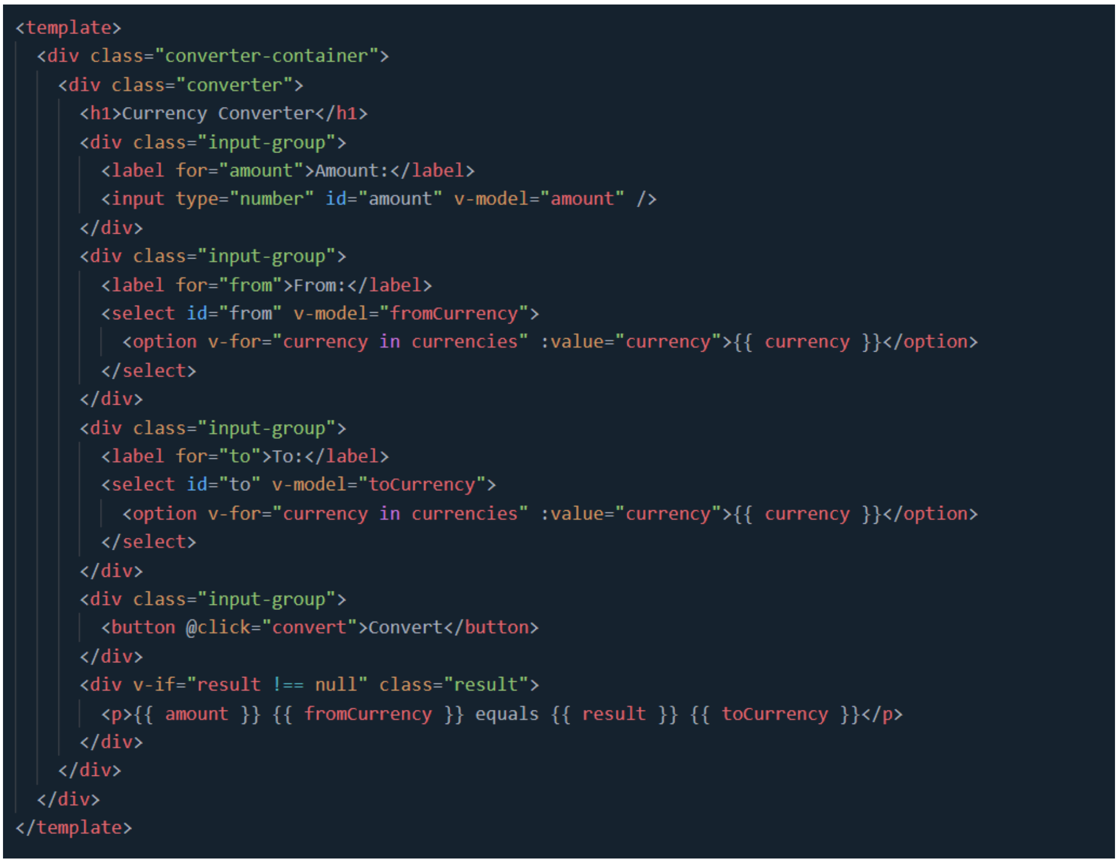
KHUSHI KALRA

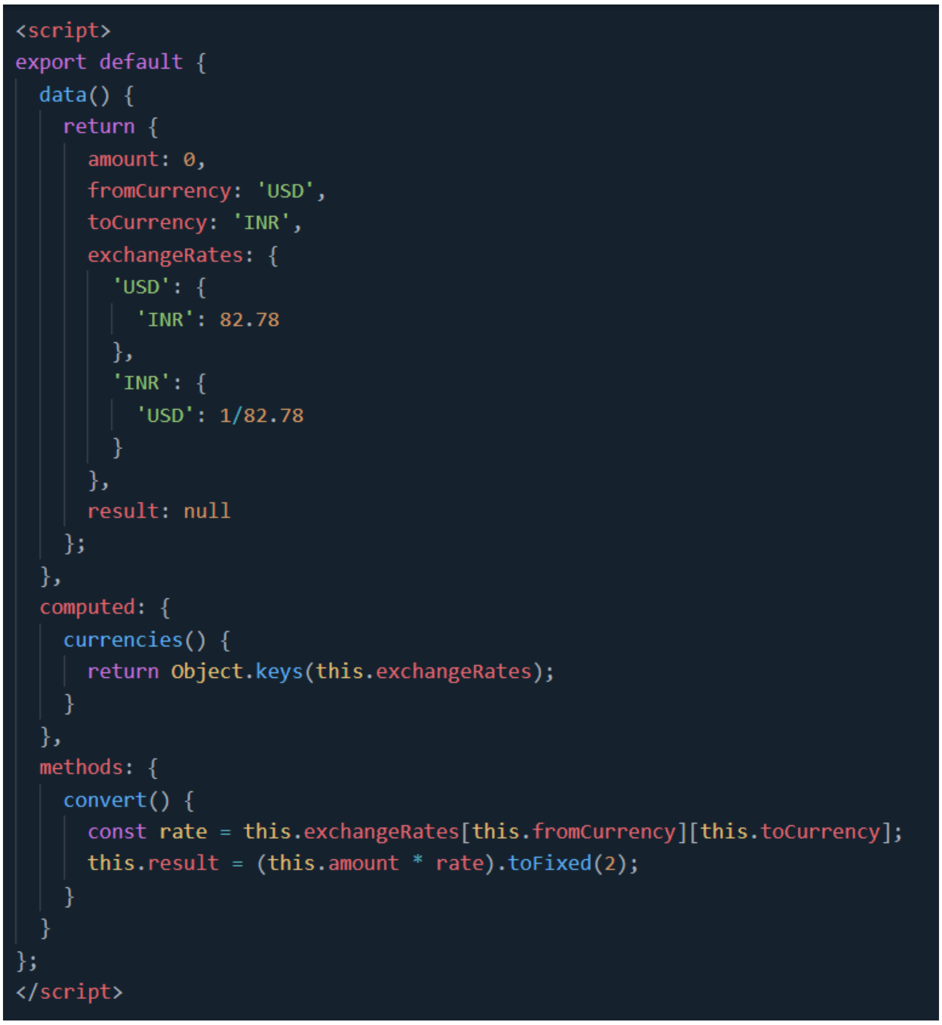
22CS3035

LAB-09

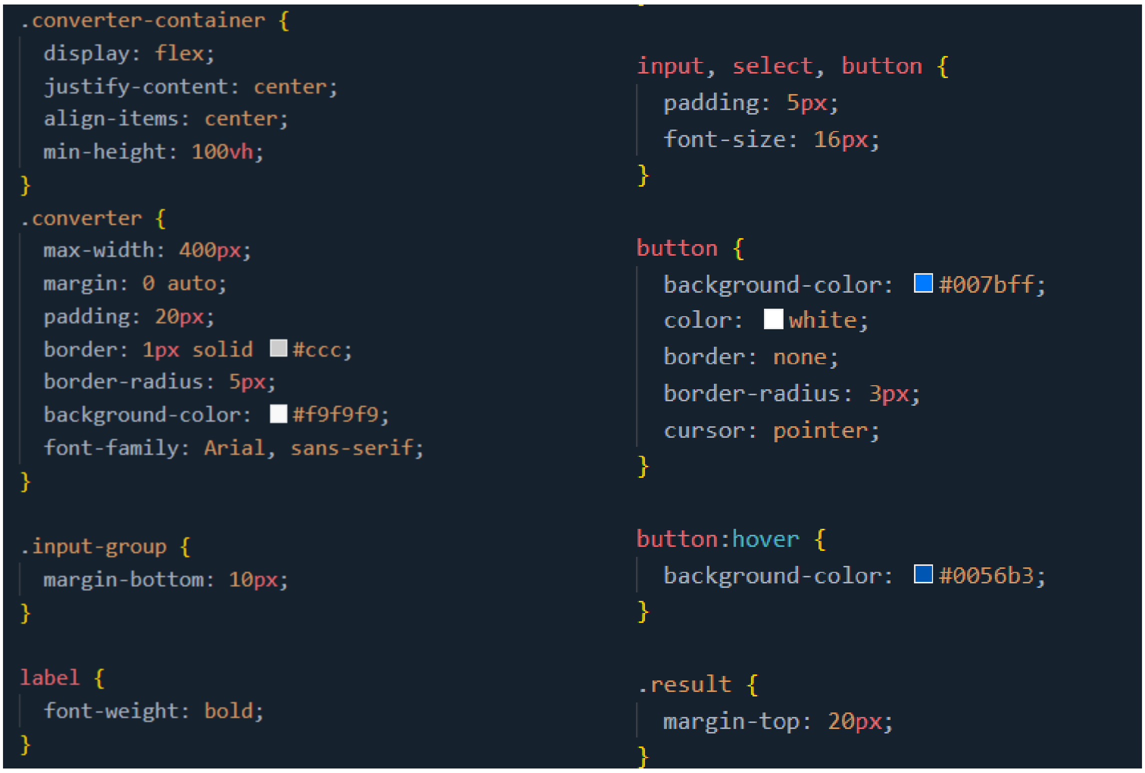
T1



```
<template>
  <div class="converter-container">
    <div class="converter">
      <h1>Currency Converter</h1>
      <div class="input-group">
        <label for="amount">Amount:</label>
        <input type="number" id="amount" v-model="amount" />
      </div>
      <div class="input-group">
        <label for="from">From:</label>
        <select id="from" v-model="fromCurrency">
          <option v-for="currency in currencies" :value="currency">{{ currency }}</option>
        </select>
      </div>
      <div class="input-group">
        <label for="to">To:</label>
        <select id="to" v-model="toCurrency">
          <option v-for="currency in currencies" :value="currency">{{ currency }}</option>
        </select>
      </div>
      <div class="input-group">
        <button @click="convert">Convert</button>
      </div>
      <div v-if="result !== null" class="result">
        <p>{{ amount }} {{ fromCurrency }} equals {{ result }} {{ toCurrency }}</p>
      </div>
    </div>
  </div>
</template>
```

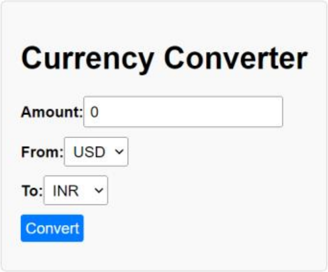


```
<script>
export default {
  data() {
    return {
      amount: 0,
      fromCurrency: 'USD',
      toCurrency: 'INR',
      exchangeRates: {
        'USD': {
          'INR': 82.78
        },
        'INR': {
          'USD': 1/82.78
        }
      },
      result: null
    };
  },
  computed: {
    currencies() {
      return Object.keys(this.exchangeRates);
    }
  },
  methods: {
    convert() {
      const rate = this.exchangeRates[this.fromCurrency][this.toCurrency];
      this.result = (this.amount * rate).toFixed(2);
    }
  }
};
</script>
```



```
.converter-container {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
}
.converter {
  max-width: 400px;
  margin: 0 auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  background-color: #f9f9f9;
  font-family: Arial, sans-serif;
}

.input-group {
  margin-bottom: 10px;
}

label {
  font-weight: bold;
}
```

```
input, select, button {
  padding: 5px;
  font-size: 16px;
}

button {
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 3px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}

.result {
  margin-top: 20px;
}
```

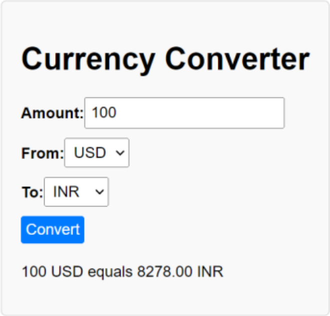


Currency Converter

Amount: 0

From: USD

To: INR



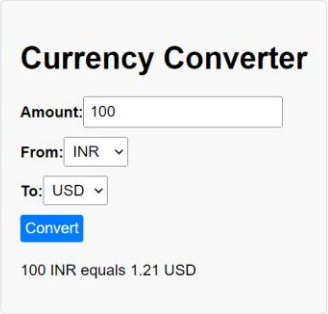
Currency Converter

Amount: 100

From: USD

To: INR

100 USD equals 8278.00 INR



Currency Converter

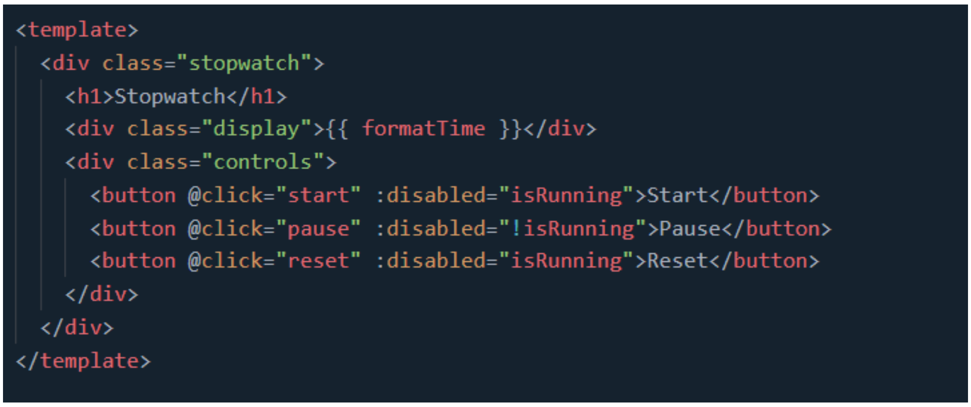
Amount: 100

From: INR

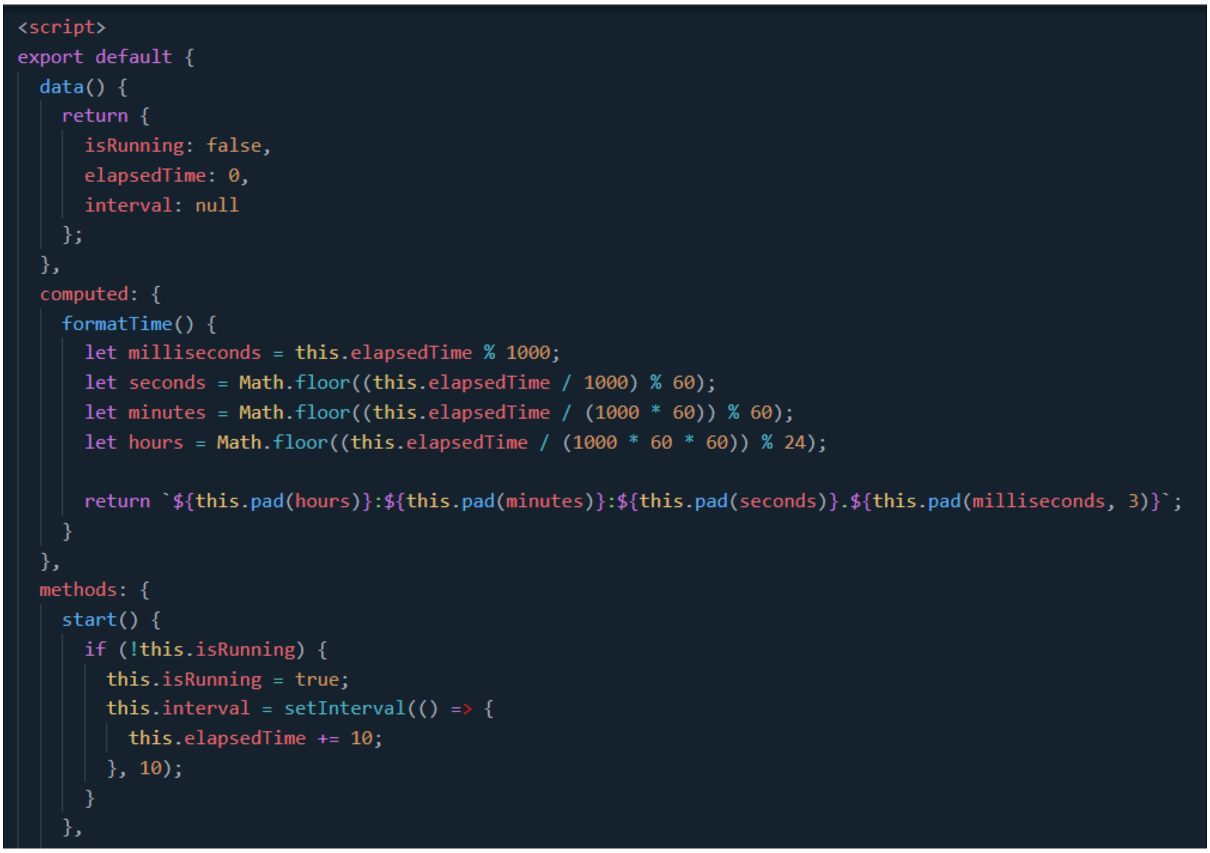
To: USD

100 INR equals 1.21 USD

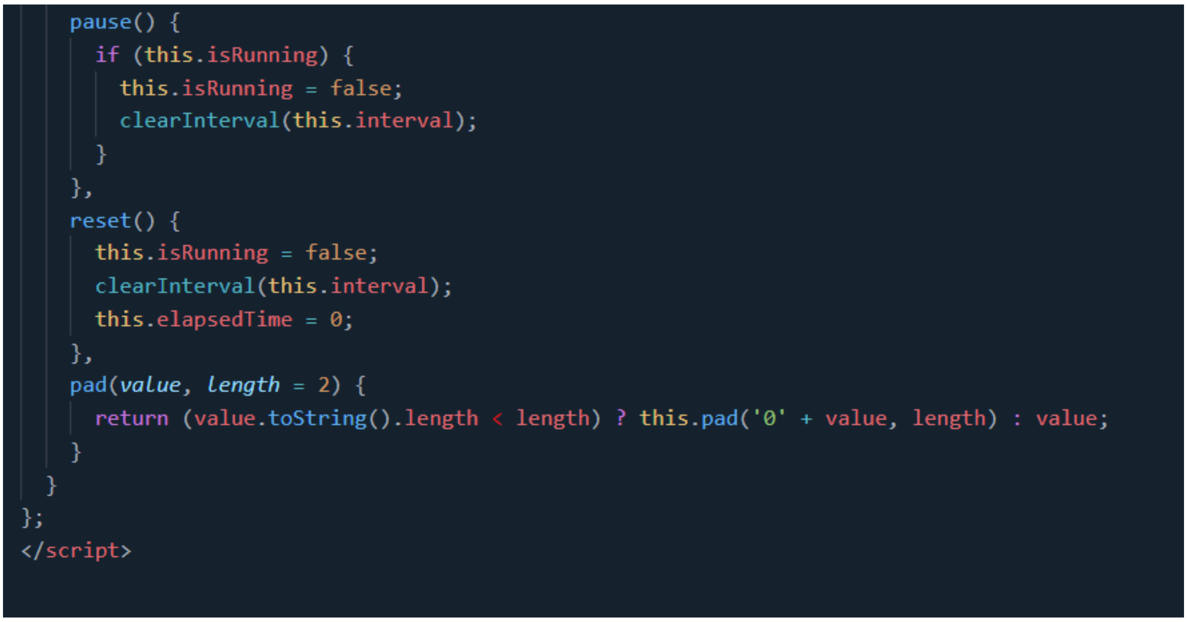
T2



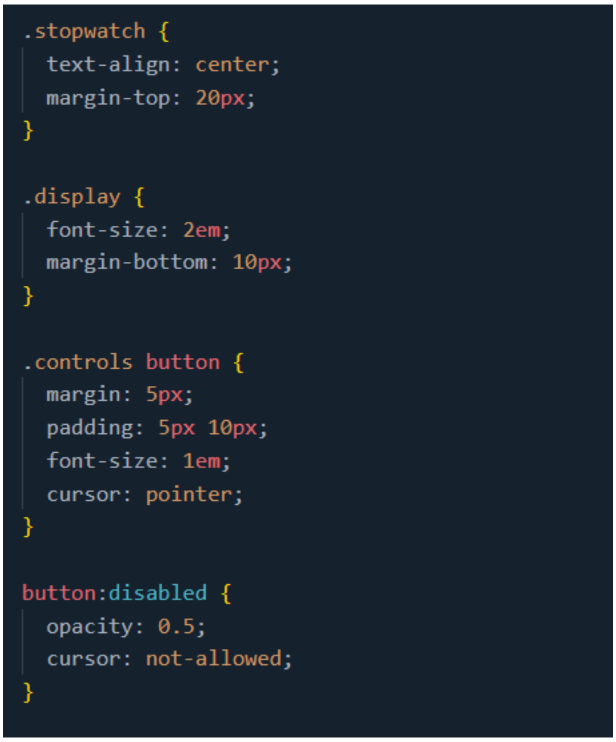
```
<template>
  <div class="stopwatch">
    <h1>Stopwatch</h1>
    <div class="display">{{ formatTime }}</div>
    <div class="controls">
      <button @click="start" :disabled="isRunning">Start</button>
      <button @click="pause" :disabled="!isRunning">Pause</button>
      <button @click="reset" :disabled="isRunning">Reset</button>
    </div>
  </div>
</template>
```



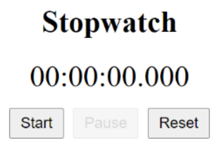
```
<script>
export default {
  data() {
    return {
      isRunning: false,
      elapsedTime: 0,
      interval: null
    };
  },
  computed: {
    formatTime() {
      let milliseconds = this.elapsedTime % 1000;
      let seconds = Math.floor((this.elapsedTime / 1000) % 60);
      let minutes = Math.floor((this.elapsedTime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

      return `${this.pad(hours)}:${this.pad(minutes)}:${this.pad(seconds)}.${this.pad(milliseconds, 3)}`;
    }
  },
  methods: {
    start() {
      if (!this.isRunning) {
        this.isRunning = true;
        this.interval = setInterval(() => {
          this.elapsedTime += 10;
        }, 10);
      }
    },
```



```
    pause() {
      if (this.isRunning) {
        this.isRunning = false;
        clearInterval(this.interval);
      }
    },
    reset() {
      this.isRunning = false;
      clearInterval(this.interval);
      this.elapsedTime = 0;
    },
    pad(value, length = 2) {
      return (value.toString().length < length) ? this.pad('0' + value, length) : value;
    }
  }
};
</script>
```

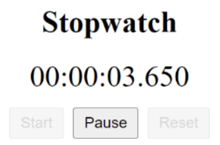


```
.stopwatch {
  text-align: center;
  margin-top: 20px;
}

.display {
  font-size: 2em;
  margin-bottom: 10px;
}

.controls button {
  margin: 5px;
  padding: 5px 10px;
  font-size: 1em;
  cursor: pointer;
}

button:disabled {
  opacity: 0.5;
  cursor: not-allowed;
}
```



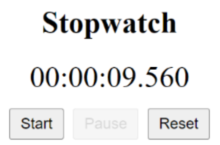
Stopwatch

00:00:00.000



Stopwatch

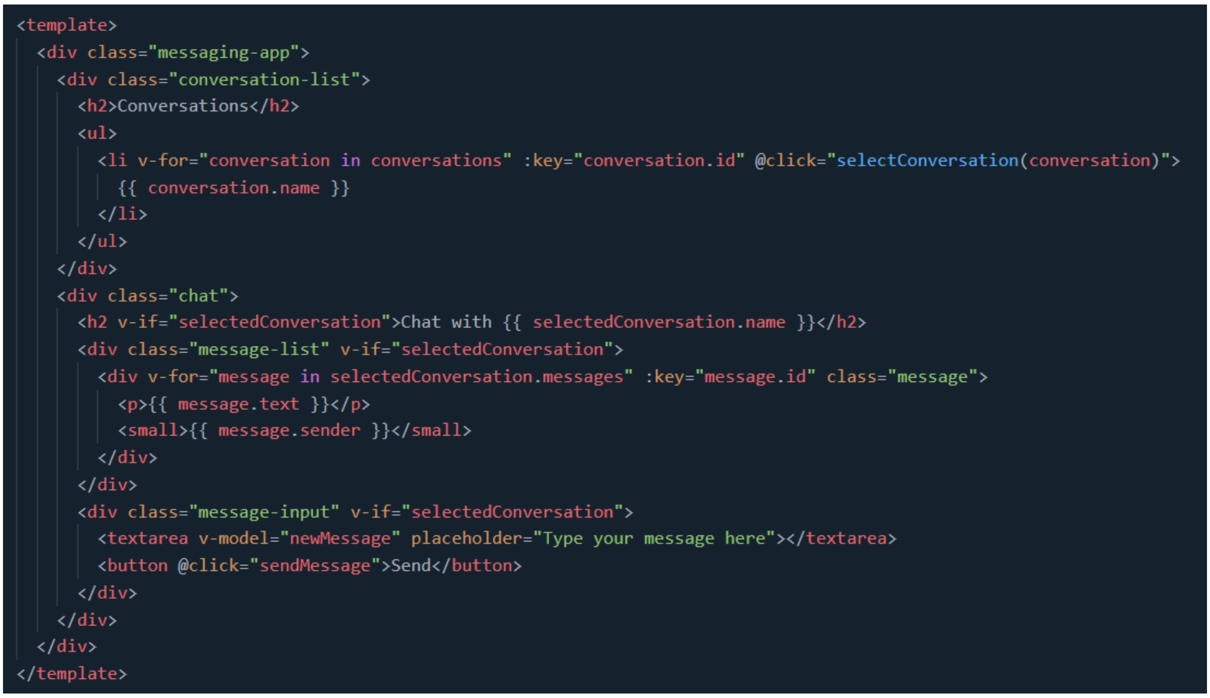
00:00:03.650



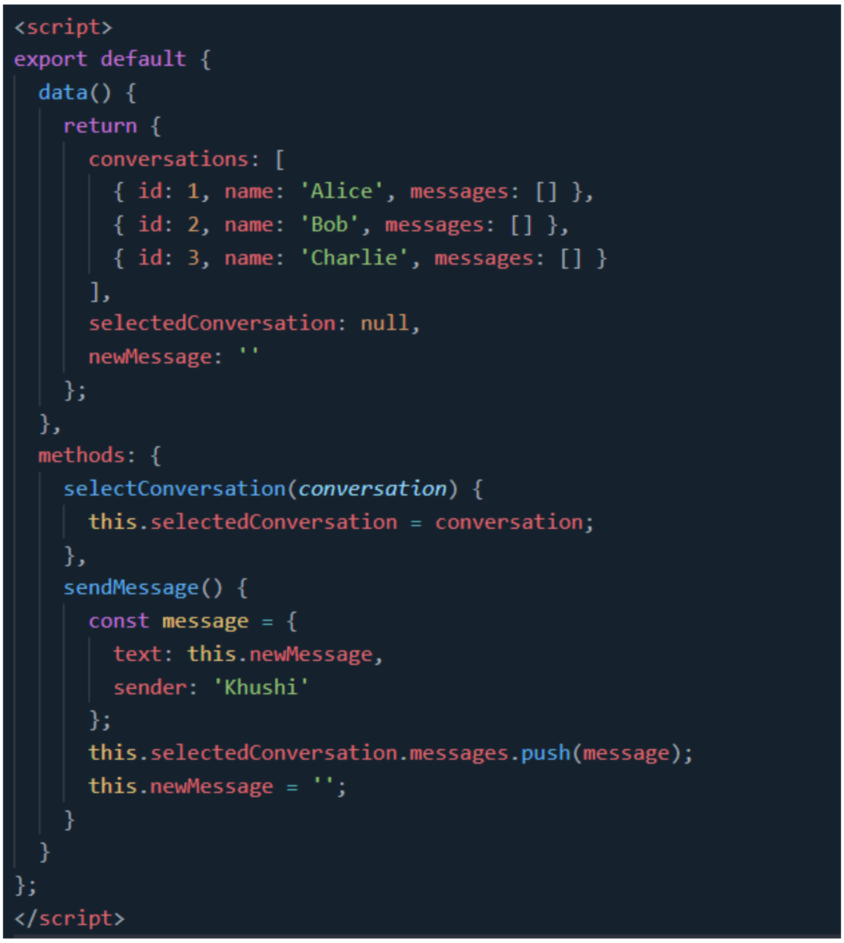
Stopwatch

00:00:09.560

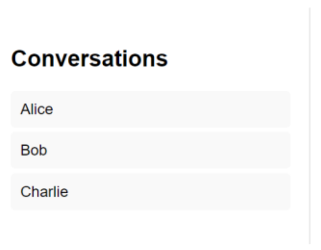
T3

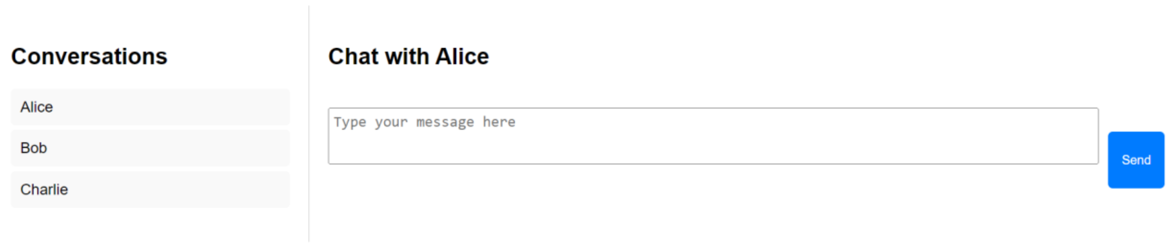


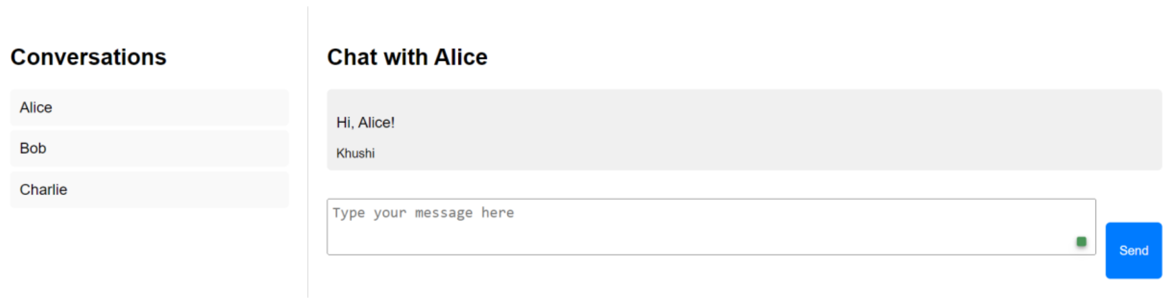
```
<template>
  <div class="messaging-app">
    <div class="conversation-list">
      <h2>Conversations</h2>
      <ul>
        <li v-for="conversation in conversations" :key="conversation.id" @click="selectConversation(conversation)">
          {{ conversation.name }}
        </li>
      </ul>
    </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: 'Khushi'
      };
      this.selectedConversation.messages.push(message);
      this.newMessage = '';
    }
  }
};
</script>
```

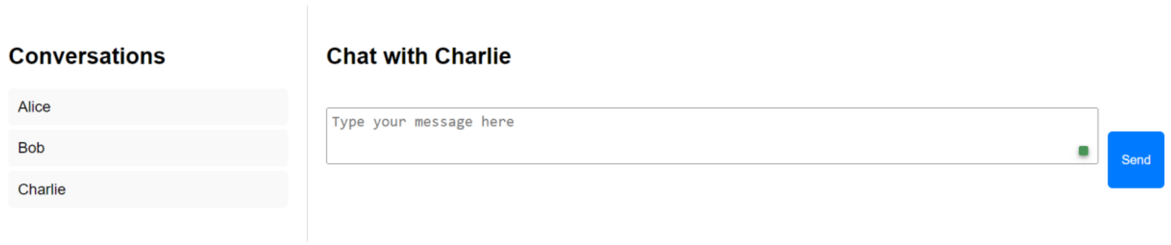






Hi, Alice!

Khushi



Conversations

- Alice
- Bob
- Charlie

Chat with Charlie