

**Name: Divyansh**

**Roll No. 22CS3025**

# **WEB Technology Assignment**

**T 1**

```
<template>

<div>

  <input type="number" v-model="amount" @input="convertCurrency">

  <select v-model="fromCurrency" @change="convertCurrency">

    <option v-for="(currency, index) in currencies" :key="index" :value="currency">{{ currency
  }}</option>

  </select>

  <p>Converted Amount: {{ convertedAmount }}</p>

  <select v-model="toCurrency" @change="convertCurrency">

    <option v-for="(currency, index) in currencies" :key="index" :value="currency">{{ currency
  }}</option>

  </select>

</div>

</template>

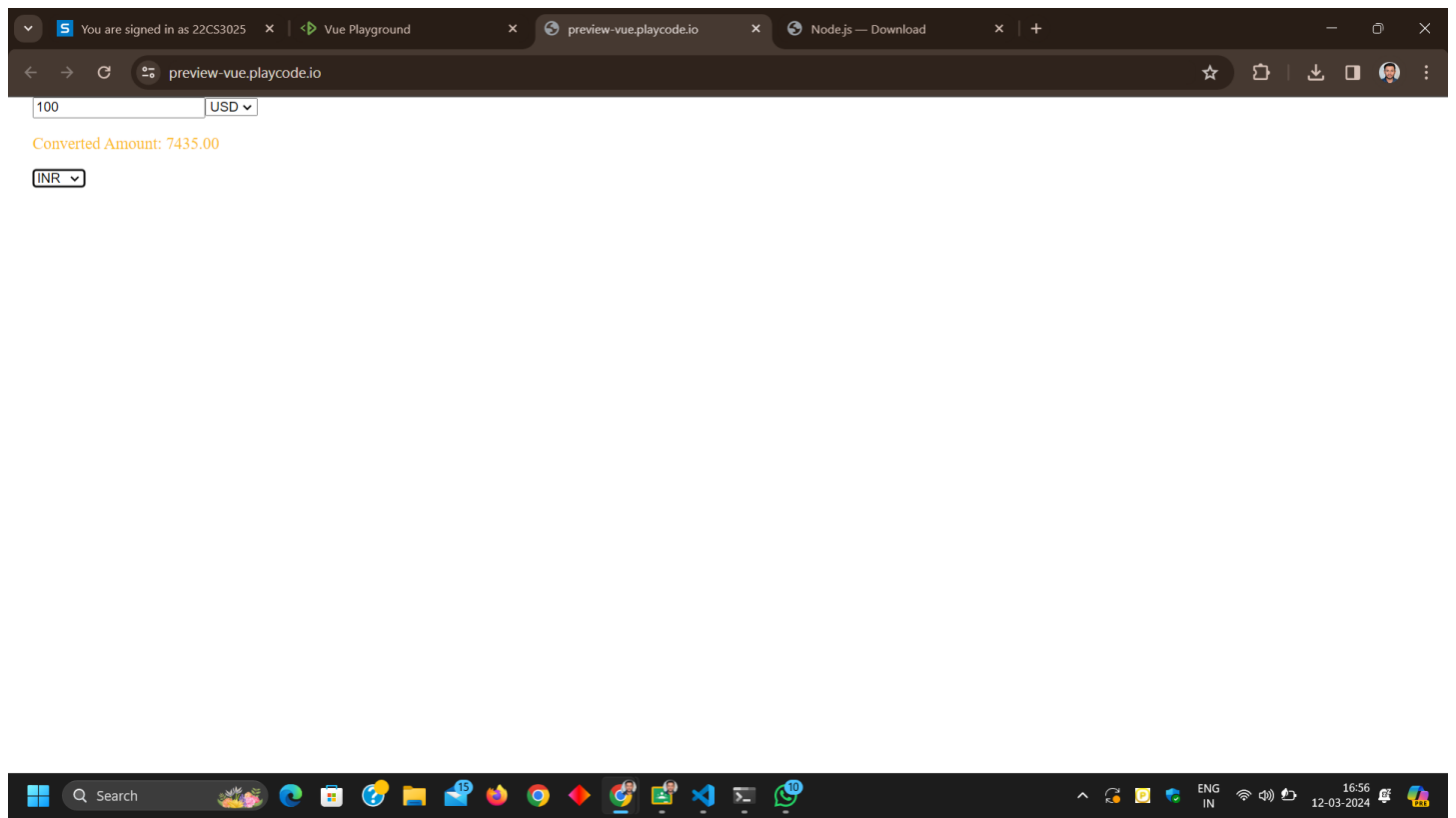
<script>

export default {

  data() {
```

```
return {  
  
  amount: 0,  
  
  fromCurrency: 'USD',  
  
  toCurrency: 'EUR',  
  
  exchangeRate: {  
  
    USD: {  
  
      EUR: 0.85,  
  
      GBP: 0.75,  
  
      INR: 74.35  
  
    },  
  
    EUR: {  
  
      USD: 1.18,  
  
      GBP: 0.88,  
  
      INR: 88.60  
  
    },  
  
    GBP: {  
  
      USD: 1.33,  
  
      EUR: 1.14,  
  
      INR: 103.45  
  
    },  
  
    INR: {  
  
      USD: 0.013,  
  
      EUR: 0.011,  
  
      GBP: 0.010  
  
    }  
  
  }  
}
```

```
    }  
  };  
},  
computed: {  
  convertedAmount() {  
    const rate = this.exchangeRate[this.fromCurrency][this.toCurrency];  
    return (this.amount * rate).toFixed(2);  
  },  
  currencies() {  
    return Object.keys(this.exchangeRate);  
  }  
},  
methods: {  
  convertCurrency() {  
  }  
}  
};  
</script>
```



## T2

```
<template>
```

```
<div>
```

```
<p>{{ formattedTime }}</p>
```

```
<button @click="startStopwatch" v-if="!running">Start</button>
```

```
<button @click="stopStopwatch" v-else>Pause</button>
```

```
<button @click="resetStopwatch">Reset</button>
```

```
</div>
```

```
</template>
```

```
<script>
```

```
export default {
```

```
data() {  
  
  return {  
  
    running: false,  
  
    time: 0  
  
  };  
  
},  
  
computed: {  
  
  formattedTime() {  
  
    const minutes = Math.floor(this.time / 60);  
  
    const seconds = this.time % 60;  
  
    return `${minutes}:${seconds < 10 ? '0' : ''}${seconds}`;  
  
  }  
  
},  
  
methods: {  
  
  startStopwatch() {  
  
    this.running = true;  
  
    this.timer = setInterval(() => {  
  
      this.time++;  
  
    }, 1000);  
  
  },  
  
  stopStopwatch() {  
  
    this.running = false;  
  
    clearInterval(this.timer);  
  
  },  
  
  resetStopwatch() {
```

```
this.time = 0;

this.running = false;

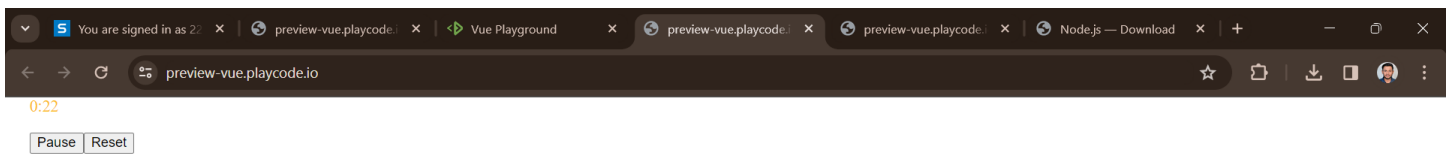
clearInterval(this.timer);

}

}

};

</script>
```



## T3

```
<template>

<div>

  <div v-if="selectedConversation">

    <h2>{{ selectedConversation.name }}</h2>

    <ul>
```

```

    <li v-for="(message, index) in selectedConversation.messages" :key="index">

      {{ message }}

    </li>

  </ul>

  <input type="text" v-model="newMessage">

  <button @click="sendMessage">Send</button>

</div>

<div v-else>

  <h2>Conversations</h2>

  <ul>

    <li v-for="(conversation, index) in conversations" :key="index"
    @click="selectConversation(conversation)">

      {{ conversation.name }}

    </li>

  </ul>

</div>

</div>

</template>

```

```

<script>

export default {

  data() {

    return {

      conversations: [

        { id: 1, name: 'Conversation 1', messages: [] },

        { id: 2, name: 'Conversation 2', messages: [] },

      ],

    }
  }
}

```

```
    { id: 3, name: 'Conversation 3', messages: [] }  
  
  ],  
  
  selectedConversation: null,  
  
  newMessage: ""  
  
};  
  
},  
  
methods: {  
  
  selectConversation(conversation) {  
  
    this.selectedConversation = conversation;  
  
  },  
  
  sendMessage() {  
  
    if (this.selectedConversation && this.newMessage.trim() !== "") {  
  
      this.selectedConversation.messages.push(this.newMessage.trim());  
  
      this.newMessage = "";  
  
      localStorage.setItem('conversations', JSON.stringify(this.conversations));  
  
    }  
  
  }  
  
},  
  
mounted() {  
  
  const storedConversations = localStorage.getItem('conversations');  
  
  if (storedConversations) {  
  
    this.conversations = JSON.parse(storedConversations);  
  
  }  
  
}  
  
};
```



&lt;/script&gt;

