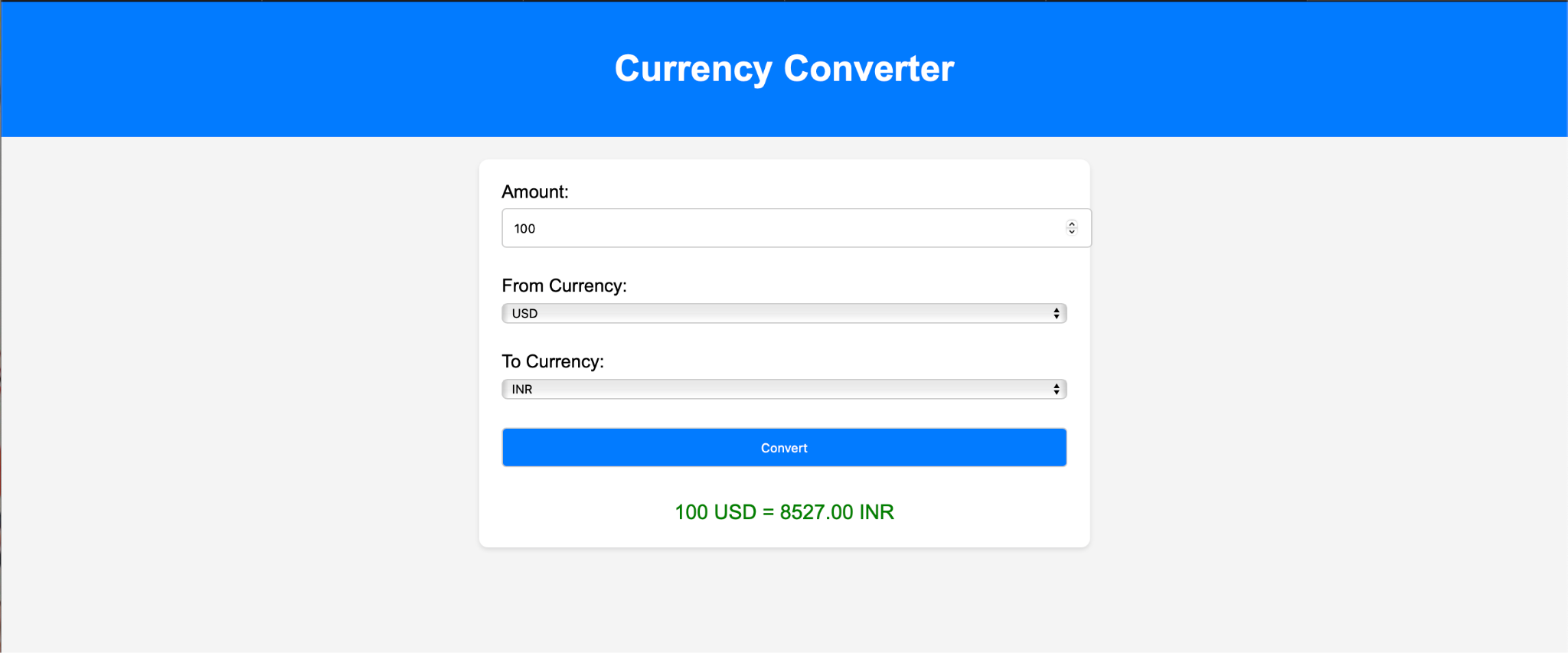
# Lab - Week 1\_A-B

PART B

|  |  |
| --- | --- |
| **Roll No.: A073** | **Name: Aryan Srivastava** |
| **Prog/Yr/Sem: MCA/FY/sem II** | **Batch: B3** |
| **Date of Experiment: 02-01-2025** | **Date of Submission: 02-01-2025** |

## Task 3:

**API- https://api.exchangerate-api.com/v4/latest/${fromCurrency}**

This application fetches real-time currency exchange rates using a free API and converts the amount entered by the user.

Features:

## Currency Converter:

* + Uses **Exchange Rate API** for real-time currency rates.
  + Converts between selected currencies and displays results.

## Weather App:

* + Uses **OpenWeather API** to fetch current weather data.
  + Displays temperature, weather description, and city information.

**Colour Code**: #007bff

**Include Learnings and Conclusion in detail.**

**CODE**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Currency Converter</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <div class="header">Currency Converter</div>

    <div class="container">

        <div class="form-group">

            <label for="amount">Amount:</label>

            <input type="number" id="amount" value="100">

        </div>

        <div class="form-group">

            <label for="fromCurrency">From Currency:</label>

            <select id="fromCurrency">

                <option value="USD">USD</option>

                <option value="EUR">EUR</option>

                <option value="INR">INR</option>

            </select>

        </div>

        <div class="form-group">

            <label for="toCurrency">To Currency:</label>

            <select id="toCurrency">

                <option value="INR">INR</option>

                <option value="USD">USD</option>

                <option value="EUR">EUR</option>

            </select>

        </div>

        <button onclick="convertCurrency()">Convert</button>

        <div id="result" class="result"></div>

    </div>

    <script>

        async function convertCurrency() {

            const amount = document.getElementById('amount').value;

            const fromCurrency = document.getElementById('fromCurrency').value;

            const toCurrency = document.getElementById('toCurrency').value;

            if (amount === '' || amount <= 0) {

                document.getElementById('result').innerHTML = "Please enter a valid amount.";

                return;

            }

            try {

                const response = await fetch(`https://api.exchangerate-api.com/v4/latest/${fromCurrency}`);

                const data = await response.json();

                const rate = data.rates[toCurrency];

                if (rate) {

                    const convertedAmount = (amount \* rate).toFixed(2);

                    document.getElementById('result').innerHTML = `${amount} ${fromCurrency} = ${convertedAmount} ${toCurrency}`;

                } else {

                    document.getElementById('result').innerHTML = "Conversion rate not available.";

                }

            } catch (error) {

                document.getElementById('result').innerHTML = "Error fetching data. Please try again later.";

                console.error("Error:", error);

            }

        }

    </script>

</body>

</html>

**Styles.css**

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f4f4f4;

}

.header {

    background-color: #007bff;

    color: white;

    text-align: center;

    padding: 15px 0;

    font-size: 24px;

}

.container {

    max-width: 400px;

    margin: 30px auto;

    background: white;

    padding: 20px;

    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);

    border-radius: 8px;

}

.form-group {

    margin-bottom: 20px;

}

label {

    display: block;

    margin-bottom: 5px;

    font-weight: bold;

}

input, select, button {

    width: 100%;

    padding: 10px;

    font-size: 16px;

    margin-bottom: 10px;

    border: 1px solid #ddd;

    border-radius: 4px;

}

button {

    background-color: #007bff;

    color: white;

    border: none;

    cursor: pointer;

}

button:hover {

    background-color: #0056b3;

}

.result {

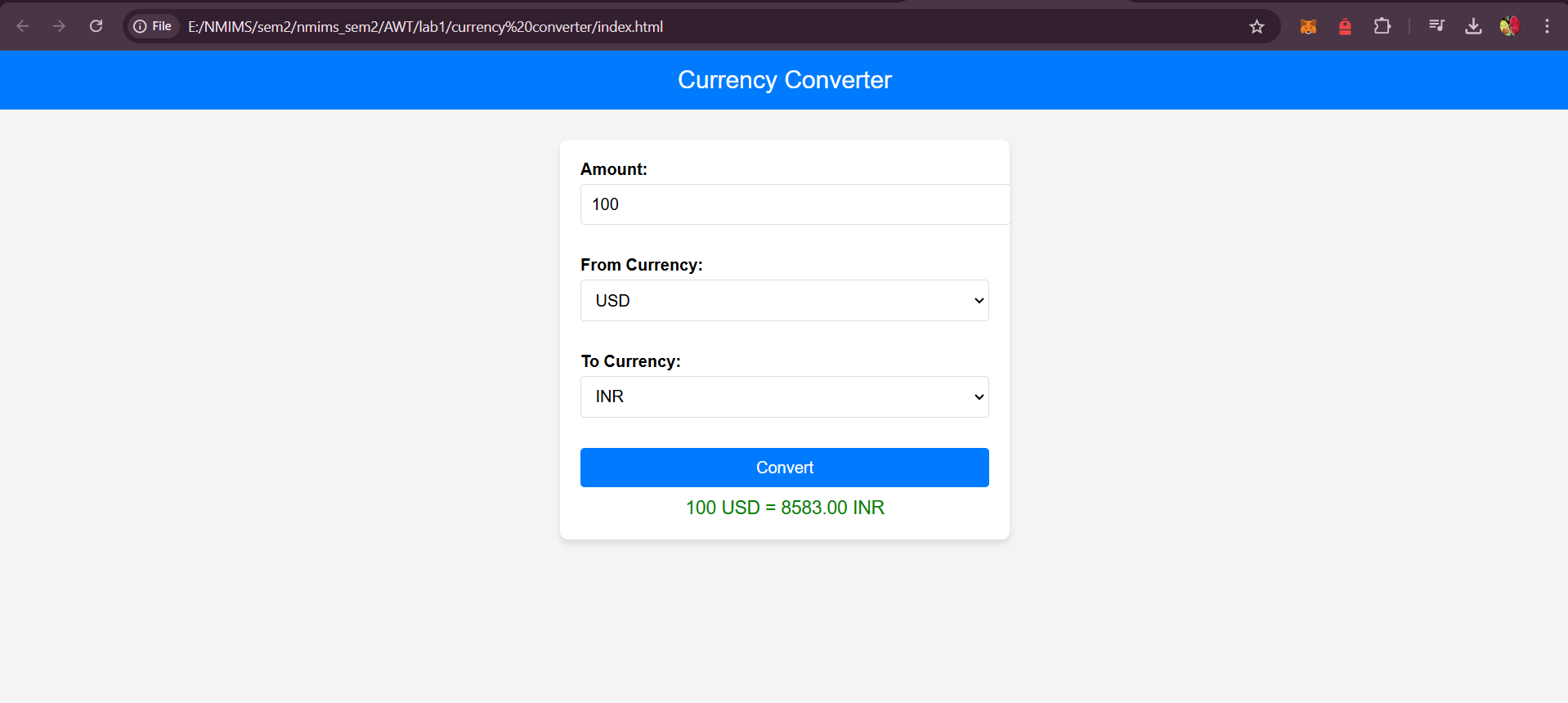
    text-align: center;

    font-size: 18px;

    color: green;

}

**OUTPUT**

****

## Task 4:

**Expense Tracker**

**CODE**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Expense Tracker</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="container">

    <h1>Expense Tracker</h1>

    <div class="input-form">

      <input type="text" id="name" placeholder="Expense Name">

      <select id="category">

        <option value="Food">Food</option>

        <option value="Transport">Transport</option>

        <option value="Shopping">Shopping</option>

        <option value="Utilities">Utilities</option>

        <option value="Other">Other</option>

      </select>

      <input type="number" id="price" placeholder="Expense Price">

      <button id="add-btn">Add Expense</button>

    </div>

    <div class="expense-list">

      <h2>Expenses:</h2>

      <table id="expenses-table">

        <thead>

          <tr>

            <th>Name</th>

            <th>Category</th>

            <th>Price</th>

          </tr>

        </thead>

        <tbody id="expenses-body">

        </tbody>

      </table>

    </div>

  </div>

  <script src="script.js"></script>

</body>

</html>

**Styles.css**

body {

    font-family: Arial, sans-serif;

    background-color: #f4f4f4;

    margin: 0;

    padding: 0;

  }

  .container {

    max-width: 600px;

    margin: 50px auto;

    background: white;

    padding: 20px;

    border-radius: 10px;

    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

  }

  h1 {

    text-align: center;

    color: #333;

  }

  .input-form {

    display: flex;

    flex-direction: column;

    gap: 10px;

  }

  .input-form input,

  .input-form select,

  .input-form button {

    padding: 10px;

    border-radius: 5px;

    border: 1px solid #ddd;

    font-size: 16px;

  }

  .input-form button {

    background-color: #007bff;

    color: white;

    border: none;

    cursor: pointer;

  }

  .input-form button:hover {

    background-color: #0056b3;

  }

  .expense-list {

    margin-top: 20px;

  }

  table {

    width: 100%;

    border-collapse: collapse;

  }

  table th, table td {

    padding: 10px;

    border: 1px solid #ddd;

    text-align: left;

  }

  table th {

    background-color: #007bff;

    color: white;

  }

**Script.js**

const addButton = document.getElementById("add-btn");

const nameInput = document.getElementById("name");

const categoryInput = document.getElementById("category");

const priceInput = document.getElementById("price");

const expensesBody = document.getElementById("expenses-body");

addButton.addEventListener("click", () => {

  const name = nameInput.value.trim();

  const category = categoryInput.value;

  const price = priceInput.value.trim();

  if (name === "" || price === "") {

    alert("Please fill out all fields.");

    return;

  }

  if (isNaN(price) || price <= 0) {

    alert("Please enter a valid price.");

    return;

  }

  const row = document.createElement("tr");

  const nameCell = document.createElement("td");

  nameCell.textContent = name;

  row.appendChild(nameCell);

  const categoryCell = document.createElement("td");

  categoryCell.textContent = category;

  row.appendChild(categoryCell);

  const priceCell = document.createElement("td");

  priceCell.textContent = `₹${parseFloat(price).toFixed(2)}`;

  row.appendChild(priceCell);

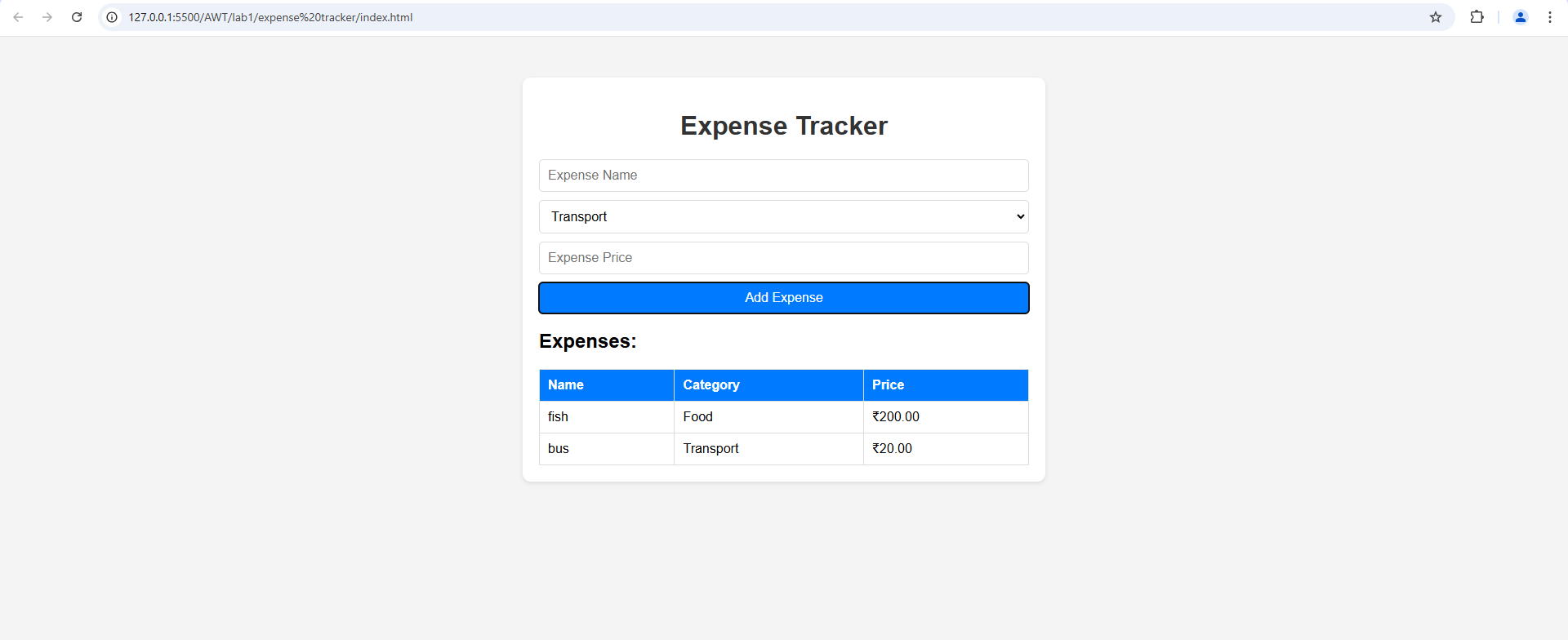
  expensesBody.appendChild(row);

  nameInput.value = "";

  priceInput.value = "";

});

**OUTPUT**

****

**Lab Deliverables:**

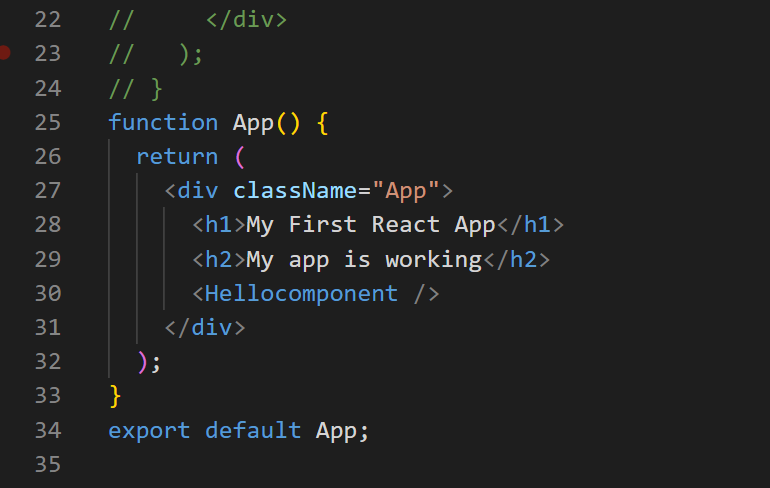
1. AReact app with a modified App.js and a new component (HelloComponent).

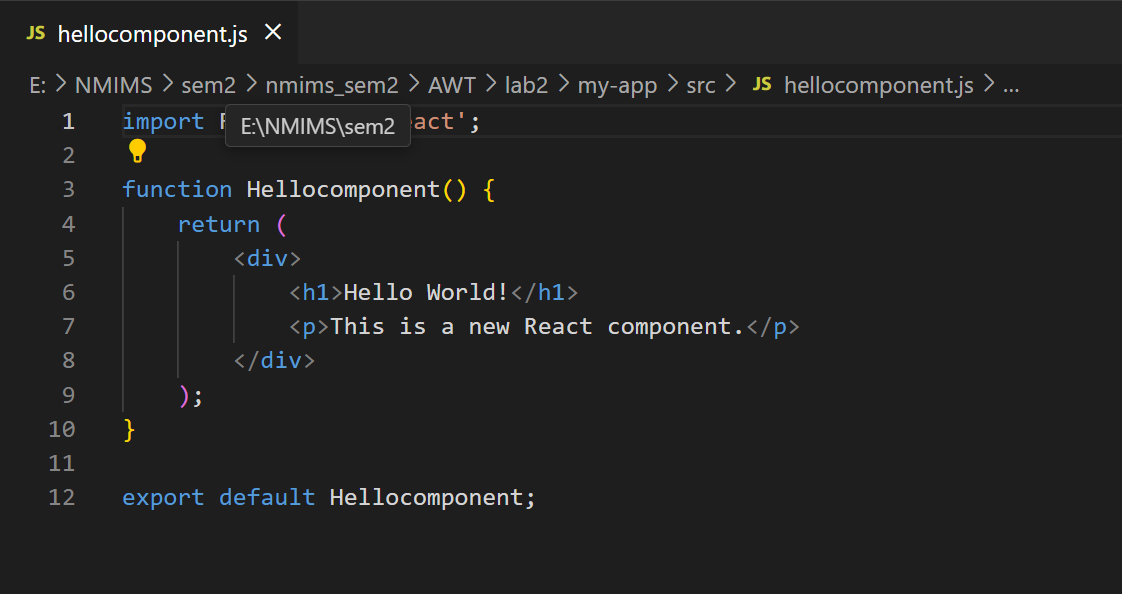
2. Screenshots of the terminal showing:

○ Successful installation of Node.js and npm.

○ Running the React server.

3. Abrowser screenshot showing the React app output.

****

****