1)pagination task:

Html code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pagination Example</title>
    <link rel="stylesheet" href="style2.css"> <!-- Link to your CSS file -->
</head>
<body>
    <!-- Container where pagination will be inserted -->
    <div id="pagination-container" class="pagination"></div>

    <!-- Script to generate pagination dynamically -->
    <script src="script.js"></script> <!-- Link to your JavaScript file -->
</body>
</html>
```

Css code:

```
.pagination {
  list-style: none;
  display: flex;
  justify-content: center;
  margin: 20px 0;
}

.pagination li {
  margin: 0 5px;
}

.pagination li a {
  padding: 8px 12px;
  text-decoration: none;
  color: #333;
  border: 1px solid #ccc;
  border-radius: 4px;
}

.pagination li a:hover {
  background-color: #f4f4f4;
}
```

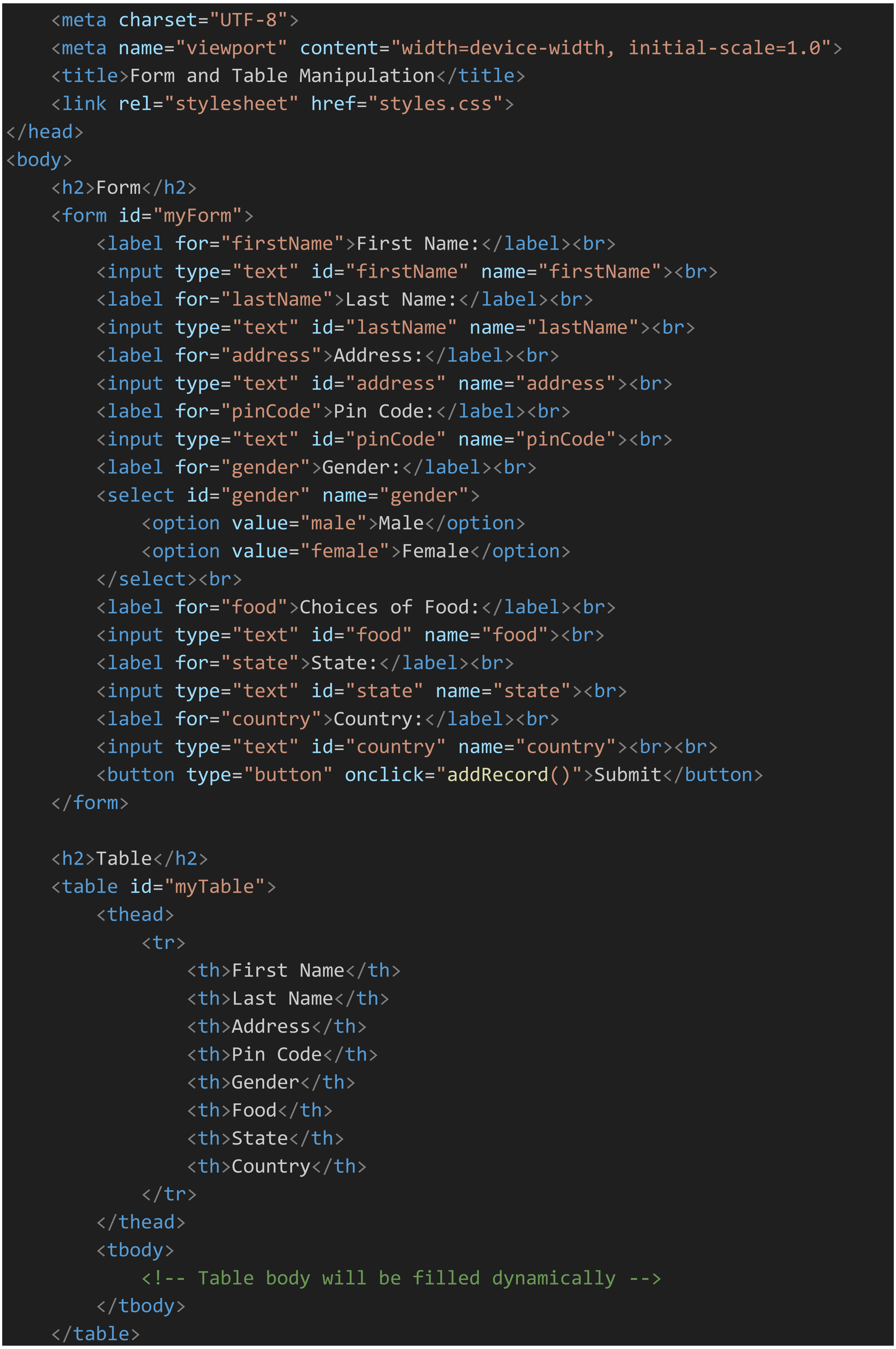
Javascript code:

```
// Select the container where pagination will be inserted
const paginationContainer = document.getElementById('pagination-container');

// Create pagination elements dynamically
const paginationList = document.createElement('ul');
paginationList.classList.add('pagination');

// Function to fetch JSON data
async function fetchData() {
    try {
        const response = await fetch("https://api.example.com/data");
        const data = await response.json();
        return data;
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

// Function to create pagination buttons
async function createPagination() {
    const totalPages = 10; // Assuming there are 10 pages
    const jsonData = await fetchData();

    for (let i = 1; i <= totalPages; i++) {
        const paginationItem = document.createElement('li');
        const paginationLink = document.createElement('a');
        paginationLink.href = '#';
        paginationLink.textContent = i;
        paginationItem.appendChild(paginationLink);
        paginationList.appendChild(paginationItem);
    }

    // Append pagination to the container
    paginationContainer.appendChild(paginationList);
}

// Call the function to create pagination
createPagination();
```

2)Html forms:

Html code:

```
<!DOCTYPE html>
<html lang="en">
<head>
```



```html
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form and Table Manipulation</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h2>Form</h2>
    <form id="myForm">
        <label for="firstName">First Name:</label><br>
        <input type="text" id="firstName" name="firstName"><br>
        <label for="lastName">Last Name:</label><br>
        <input type="text" id="lastName" name="lastName"><br>
        <label for="address">Address:</label><br>
        <input type="text" id="address" name="address"><br>
        <label for="pinCode">Pin Code:</label><br>
        <input type="text" id="pinCode" name="pinCode"><br>
        <label for="gender">Gender:</label><br>
        <select id="gender" name="gender">
            <option value="male">Male</option>
            <option value="female">Female</option>
        </select><br>
        <label for="food">Choices of Food:</label><br>
        <input type="text" id="food" name="food"><br>
        <label for="state">State:</label><br>
        <input type="text" id="state" name="state"><br>
        <label for="country">Country:</label><br>
        <input type="text" id="country" name="country"><br><br>
        <button type="button" onclick="addRecord()">Submit</button>
    </form>

    <h2>Table</h2>
    <table id="myTable">
        <thead>
            <tr>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Address</th>
                <th>Pin Code</th>
                <th>Gender</th>
                <th>Food</th>
                <th>State</th>
                <th>Country</th>
            </tr>
        </thead>
        <tbody>
            <!-- Table body will be filled dynamically -->
        </tbody>
    </table>
    <script src="script.js"></script>
</body>
</html>
```

Css code:

```
table {
  border-collapse: collapse;
  width: 100%;
}
th, td {
  border: 1px solid #dddddd;
  text-align: left;
  padding: 8px;
}
th {
  background-color: #f2f2f2;
}
```

Javascript code:

```
function addRecord() {
    var formData = {};
    var form = document.getElementById("myForm").elements;
    for (var i = 0; i < form.length; i++) {
        if (form[i].type !== "button") {
            formData[form[i].name] = form[i].value;
        }
    }
    var tableRef =
document.getElementById("myTable").getElementsByTagName('tbody')[0];
    var newRow = tableRef.insertRow();
    for (var key in formData) {
        var newCell = newRow.insertCell();
        var newText = document.createTextNode(formData[key]);
        newCell.appendChild(newText);
    }
    clearForm();
}

function clearForm() {
    var form = document.getElementById("myForm").elements;
    for (var i = 0; i < form.length; i++) {
        if (form[i].type !== "button") {
            form[i].value = "";
        }
    }
}
```