

Project Report: Real-time Water Consumption Data Sharing Platform

Table of Contents

| | |
|-----|--|
| 1. | Project Overview |
| 2. | Objectives |
| 3. | Technology Stack |
| 4. | System Architecture |
| 5. | Front-End Development <ul style="list-style-type: none">HTML StructureCSS StylingJavaScript for Real-Time Data Visualization |
| 6. | Back-End Development <ul style="list-style-type: none">Server SetupDatabase ConfigurationUser AuthenticationAPI DevelopmentData Storage and RetrievalReal-Time CapabilitiesSecurity Measures |
| 7. | IoT Sensor Integration |
| 8. | Testing and Quality Assurance |
| 9. | Deployment |
| 10. | User Engagement Features |
| 11. | Monitoring and Maintenance |
| 12. | Feedback and Improvement |
| 13. | Marketing and Promotion |
| 14. | Conclusion |

5. Front-End Development

5.1 HTML Structure

```
<!DOCTYPE html>

<html>

<head>

  <title>Water Consumption Data Platform</title>

</head>

<body>

  <header>
```

```
    <h1>Real-time Water Consumption Data</h1>
</header>

<section id="data-visualization">
    <!-- Real-time data visualization here -->
</section>

<section id="conservation-tips">
    <!-- Water conservation tips section -->
</section>

<section id="user-interactions">
    <!-- Filters, search, and notifications here -->
</section>
</body>
</html>
```

5.2 CSS Styling

```
/* Styles for the header */
header {
    background-color: #007BFF;
    color: #fff;
    text-align: center;
    padding: 10px;
}

/* Styles for data visualization section */
#data-visualization {
    /* Your styles here */
}

/* Styles for conservation tips section */
#conservation-tips {
    /* Your styles here */
}
```

```
}

/* Styles for user interactions section */
#user-interactions {
  /* Your styles here */
}
```

5.3 JavaScript for Real-Time Data Visualization

```
// Sample code to fetch real-time data and update the visualization
const dataVisualization = document.getElementById('data-visualization');

function updateDataVisualization(data) {
  // Code to update the data visualization (e.g., using Chart.js or D3.js)
}

// Example: Fetch data from an API
fetch('/api/realtime-data')
  .then(response => response.json())
  .then(data => {
    updateDataVisualization(data);
  });
```