Attendance Management Website - Project Report

Abstract

This project implements a lightweight web-based Attendance Management System suitable for use on mobile devices and desktops. It allows teachers to mark attendance for 100 students, edit student names, save and load records from the browser (LocalStorage), export attendance as CSV, and trigger an SMS app with a pre-filled absent message for parents.

Objectives

- Provide an easy-to-use interface for marking attendance. - Auto-generate 100 student rows with roll numbers. - Allow editing student names and phone numbers. - Save attendance data locally using LocalStorage. - Export attendance records as CSV. - Trigger SMS compose with pre-filled message when marking absent (opens device SMS app).

Tools & Technologies

- Frontend: HTML, CSS, JavaScript - Storage: Browser LocalStorage (no server required) - Platform: Any modern web browser (mobile recommended) - Optional: Replit or Spck Editor for mobile development

System Design & Workflow

1. On first load the app generates a table of 100 students (Roll No. 1–100) with editable name and phone fields. 2. Teacher can mark each student Present / Absent. The status is saved to LocalStorage. 3. When marking Absent the app opens the device's SMS composer with a prefilled message. The teacher then sends the SMS. 4. The teacher can save/load attendance, clear data, and export records to CSV for offline records.

How to Run (on Mobile)

1. Using a code editor app (Spck Editor) or Replit, create a new HTML project and paste the 'index.html' code. 2. Open the file in the mobile browser (Chrome). The app runs entirely in the browser—no hosting required. 3. Edit names/phone numbers, mark attendance, and use buttons to save or export.

Files Included

- index.html (contains HTML, CSS, JS) - (All code is embedded in index.html for simplicity)

Code: index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <title>Attendance System</title>
  <style>
    body { font-family: Arial, sans-serif; background:#f4f4f4; padding:10px; }
    .container { max-width:980px; margin:auto; background:#fff; padding:12px; border-radius:8px; box-shadow:0 2
    table { width:100%; border-collapse: collapse; margin-top:10px; font-size:13px; }
    th, td { border:1px solid #ddd; padding:8px; text-align:center; vertical-align:middle; }
    th { background:#f0f0f0; }
    input[type="text"], input[type="tel"] { width:95%; padding:6px; border-radius:4px; border:1px solid #ccc; b
    button { padding:6px 10px; margin:2px; border:none; border-radius:6px; cursor:pointer; }
    .controls { display:flex; gap:8px; flex-wrap:wrap; margin-top:8px; }
    .present { background:green; color:white; }
    .absent { background:red; color:white; }
    .small { font-size:12px; padding:5px 8px; }
    @media (max-width:600px) {
      table, th, td { font-size:12px; }
      .container { padding:8px; }
    }
  </style>
</head>
<body>
  <div class="container">
```

```
  <h2>Attendance Management System (100 Students)</h2>
  <div class="controls">
    <button onclick="saveAttendance()" class="small">Save</button>
    <button onclick="loadAttendance()" class="small">Load</button>
    <button onclick="clearAttendance()" class="small">Clear</button>
    <button onclick="exportCSV()" class="small">Export CSV</button>
    <button onclick="markAllPresent()" class="small">Mark All Present</button>
    <button onclick="markAllAbsent()" class="small">Mark All Absent (No SMS)</button>
  </div>

  <table id="attendanceTable">
    <thead>
      <tr>
        <th>Roll No</th>
        <th>Name</th>
        <th>Phone</th>
        <th>Status</th>
      </tr>
    </thead>
    <tbody></tbody>
  </table>
  <p class="small">Note: Marking "Absent" opens your device SMS app with a prefilled message; you must send t
</div>

<script>
  const STORAGE_KEY = "attendance_v1";
  const NUM_STUDENTS = 100;

  // Utility to create default phone number (change as needed)
  function defaultPhone(i) {
    // Example: 999000001 ... adjust as required
    const base = "999000";
    return base + String(i).padStart(3, '0');
  }

  // Initialize table
  function initTable() {
    const tbody = document.querySelector("#attendanceTable tbody");
    tbody.innerHTML = "";
    for (let i = 1; i <= NUM_STUDENTS; i++) {
      const tr = document.createElement("tr");
      tr.innerHTML = `
        <td>${i}</td>
        <td><input type="text" value="Student ${i}" id="name_${i}" /></td>
        <td><input type="tel" value="${defaultPhone(i)}" id="phone_${i}" /></td>
        <td id="status_${i}">
          <button onclick="mark(${i}, 'present')" class="small present">Present</button>
          <button onclick="mark(${i}, 'absent')" class="small absent">Absent</button>
        </td>
      `;
      tbody.appendChild(tr);
    }
  }

  // Mark attendance
  function mark(roll, status) {
    const statusCell = document.getElementById(`status_${roll}`);
    if (status === "present") {
      statusCell.innerHTML = "PRESENT";
      statusCell.style.color = "green";
    } else {
      statusCell.innerHTML = "ABSENT";
      statusCell.style.color = "red";
      const name = document.getElementById(`name_${roll}`).value || `Student ${roll}`;
      const phone = document.getElementById(`phone_${roll}`).value || defaultPhone(roll);
      const message = `Student ${name} (Roll No: ${roll}) is ABSENT today.`;
      // open SMS composer (works on most mobile devices)
      window.location.href = `sms:${phone}?body=${encodeURIComponent(message)}`;
    }
  }

  // Save to LocalStorage
  function saveAttendance() {
    const data = [];
    for (let i = 1; i <= NUM_STUDENTS; i++) {
      const name = document.getElementById(`name_${i}`).value;
      const phone = document.getElementById(`phone_${i}`).value;
      const statusCell = document.getElementById(`status_${i}`);
```

```
    const status = statusCell ? statusCell.textContent.trim() : "";
    data.push({ roll: i, name, phone, status });
  }
  localStorage.setItem(STORAGE_KEY, JSON.stringify(data));
  alert("Attendance saved to LocalStorage.");
}

// Load from LocalStorage
function loadAttendance() {
  const raw = localStorage.getItem(STORAGE_KEY);
  if (!raw) { alert("No saved attendance found."); return; }
  const data = JSON.parse(raw);
  data.forEach(item => {
    document.getElementById(`name_${item.roll}`).value = item.name;
    document.getElementById(`phone_${item.roll}`).value = item.phone;
    const statusCell = document.getElementById(`status_${item.roll}`);
    if (item.status === "PRESENT") {
      statusCell.innerHTML = "PRESENT";
      statusCell.style.color = "green";
    } else if (item.status === "ABSENT") {
      statusCell.innerHTML = "ABSENT";
      statusCell.style.color = "red";
    } else {
      statusCell.innerHTML = `
        <button onclick="mark(${item.roll}, 'present')" class="small present">Present</button>
        <button onclick="mark(${item.roll}, 'absent')" class="small absent">Absent</button>
      `;
      statusCell.style.color = "black";
    }
  });
  alert("Attendance loaded from LocalStorage.");
}

// Clear attendance statuses (keeps names & phones)
function clearAttendance() {
  if (!confirm("Clear all attendance statuses?")) return;
  for (let i = 1; i <= NUM_STUDENTS; i++) {
    const statusCell = document.getElementById(`status_${i}`);
    statusCell.innerHTML = `
      <button onclick="mark(${i}, 'present')" class="small present">Present</button>
      <button onclick="mark(${i}, 'absent')" class="small absent">Absent</button>
    `;
    statusCell.style.color = "black";
  }
  localStorage.removeItem(STORAGE_KEY);
  alert("Attendance cleared.");
}

// Export CSV
function exportCSV() {
  let csv = "Roll,Name,Phone,Status\n";
  for (let i = 1; i <= NUM_STUDENTS; i++) {
    const name = document.getElementById(`name_${i}`).value.replace(/,/g, "");
    const phone = document.getElementById(`phone_${i}`).value.replace(/,/g, "");
    const status = document.getElementById(`status_${i}`).textContent.trim();
    csv += `${i},${name},${phone},${status}\n`;
  }
  const blob = new Blob([csv], { type: "text/csv" });
  const url = URL.createObjectURL(blob);
  const a = document.createElement("a");
  a.href = url;
  a.download = "attendance.csv";
  document.body.appendChild(a);
  a.click();
  document.body.removeChild(a);
  URL.revokeObjectURL(url);
}

// Bulk actions
function markAllPresent() {
  for (let i = 1; i <= NUM_STUDENTS; i++) {
    const statusCell = document.getElementById(`status_${i}`);
    statusCell.innerHTML = "PRESENT";
    statusCell.style.color = "green";
  }
}
function markAllAbsent() {
  if (!confirm("This will mark all as ABSENT without opening SMS. Continue?")) return;
```

```
      for (let i = 1; i <= NUM_STUDENTS; i++) {
        const statusCell = document.getElementById(`status_${i}`);
        statusCell.innerHTML = "ABSENT";
        statusCell.style.color = "red";
      }
    }

    // Init on load
    window.addEventListener("load", () => {
      initTable();
      // Try to load previous data automatically if exists
      if (localStorage.getItem(STORAGE_KEY)) {
        // do not auto-load to avoid unexpected overwrites; user can choose Load
        console.log("Saved attendance found. Click 'Load' to restore.");
      }
    });
  </script>
</body>
</html>
```

Testing & Output

Tested on mobile browsers (Chrome) and desktop browsers. Features verified: - 100 student rows generation - Editing names and phone numbers - Mark Present/Absent - Save/Load using LocalStorage - Export CSV - SMS composer opening on Absent button (mobile)

Future Enhancements

- Add server-side backend (Node.js/PHP/Flask) and database (MySQL/SQLite) for permanent storage. - Integrate SMS API (Twilio) to send SMS automatically from server. - Add authentication for teachers and role-based access. - Add reports, charts and attendance percentage calculations.

Conclusion

This mini-project delivers a complete, mobile-friendly attendance system that requires no backend and can be used as a prototype or as a submission for academic evaluation. The code is simple, modular, and easy to extend.

How to Submit / Demo

1. Zip the 'index.html' and upload it to your project submission portal OR 2. Host on Replit and provide the project link OR 3. Demonstrate on mobile: open index.html in browser, mark attendance, show export CSV and SMS flow.