Attendance Management Website - Code Bundle

Description:

This PDF contains a simple attendance management website implementation suitable for running on a PHP-enabled server. Frontend: HTML/CSS/JavaScript. Backend: single PHP file using SQLite for storage. Files included below and instructions to get started.

Files included:

1) index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Attendance Management</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Attendance Management</h1>

    <section id="students">
      <h2>Students</h2>
      <form id="addStudentForm">
        <input id="name" placeholder="Student name" required>
        <input id="roll" placeholder="Roll number" required>
        <button type="submit">Add student</button>
      </form>
      <table id="studentTable">
        <thead><tr><th>Roll</th><th>Name</th><th>Mark</th></tr></thead>
        <tbody></tbody>
      </table>
    </section>

    <section id="attendance">
      <h2>Attendance</h2>
      <label>Date: <input type="date" id="attDate"></label>
      <button id="saveAttendance">Save Attendance</button>
      <div id="message"></div>
    </section>

    <section id="reports">
      <h2>Report</h2>
      <label>Date: <input type="date" id="reportDate"></label>
      <button id="loadReport">Load</button>
      <div id="reportArea"></div>
    </section>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

2) styles.css

```
*{box-sizing:border-box;font-family:Arial,Helvetica,sans-serif}
body{background:#f6f8fa;padding:20px}
.container{max-width:900px;margin:0 auto;background:#fff;padding:20px;border-radius:8px;box-shadow:0 4px
h1,h2{color:#111;margin:8px 0}
table{width:100%;border-collapse:collapse;margin-top:10px}
table,th,td{border:1px solid #ddd}
th,td{padding:8px;text-align:left}
button{padding:6px 10px;border:none;background:#0b74de;color:#fff;border-radius:4px;cursor:pointer}
input{padding:6px;margin-right:6px}
```

3) script.js

```
const api = 'api.php';

async function fetchStudents(){
  const res = await fetch(api+'?action=list_students');
  const data = await res.json();
  const tbody = document.querySelector('#studentTable tbody');
  tbody.innerHTML='';
  data.forEach(s => {
    const tr = document.createElement('tr');
    tr.innerHTML = `<td>${s.roll}</td><td>${s.name}</td><td><input type="checkbox" data-roll="${s.roll}">
    tbody.appendChild(tr);
  });
}

document.getElementById('addStudentForm').addEventListener('submit', async (e)=>{
  e.preventDefault();
  const name = document.getElementById('name').value.trim();
  const roll = document.getElementById('roll').value.trim();
  if(!name||!roll) return;
  await fetch(api, {method:'POST', headers:{'Content-Type':'application/json'}, body:JSON.stringify({acti
  document.getElementById('name').value=''; document.getElementById('roll').value='';
  fetchStudents();
});

document.getElementById('saveAttendance').addEventListener('click', async ()=>{
  const date = document.getElementById('attDate').value;
  if(!date){ alert('Choose a date'); return; }
  const checks = Array.from(document.querySelectorAll('#studentTable tbody input[type=checkbox]'));
  const present = checks.filter(c=>c.checked).map(c=>c.dataset.roll);
  await fetch(api, {method:'POST', headers:{'Content-Type':'application/json'}, body:JSON.stringify({acti
  document.getElementById('message').textContent = 'Attendance saved.';
});

document.getElementById('loadReport').addEventListener('click', async ()=>{
  const date = document.getElementById('reportDate').value;
  if(!date){ alert('Choose a date'); return; }
  const res = await fetch(api+'?action=get_attendance&date='+encodeURIComponent(date));
  const data = await res.json();
  const area = document.getElementById('reportArea');
  if(!data){ area.innerHTML = 'No attendance for this date.'; return; }
  area.innerHTML = '<h3>Present Students</h3><ul>' + data.present.map(r=>`<li>${r.roll} - ${r.name}</li>`
});

// initial load
fetchStudents();
```

4) api.php (PHP backend using SQLite)

```php
<?php
// Simple API using SQLite. Save this as api.php and ensure PHP has PDO SQLite enabled.
// It will create attendance.db in the same folder automatically.
$dbfile = __DIR__ . '/attendance.db';
$pdo = new PDO('sqlite:' . $dbfile);
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// initialize tables
$pdo->exec("CREATE TABLE IF NOT EXISTS students (id INTEGER PRIMARY KEY AUTOINCREMENT, roll TEXT UNIQUE,
$pdo->exec("CREATE TABLE IF NOT EXISTS attendance (id INTEGER PRIMARY KEY AUTOINCREMENT, student_id INTEG

$method = $_SERVER['REQUEST_METHOD'];
$action = null;
if($method === 'GET'){
    $action = $_GET['action'] ?? null;
} else {
    $payload = json_decode(file_get_contents('php://input'), true);
    $action = $payload['action'] ?? null;
}

header('Content-Type: application/json');

try {
    if($action === 'list_students'){
        $stmt = $pdo->query('SELECT id, roll, name FROM students ORDER BY roll');
        $students = $stmt->fetchAll(PDO::FETCH_ASSOC);
        echo json_encode($students);
        exit;
    } elseif($action === 'add_student'){
        $name = $payload['name'] ?? '';
        $roll = $payload['roll'] ?? '';
        $stmt = $pdo->prepare('INSERT OR IGNORE INTO students (roll,name) VALUES (?,?)');
        $stmt->execute([$roll, $name]);
        echo json_encode(['ok'=>true]);
        exit;
    } elseif($action === 'save_attendance'){
        $date = $payload['date'];
        $present = $payload['present'] ?? [];
        // remove existing marks for date
        $pdo->prepare('DELETE FROM attendance WHERE date = ?')->execute([$date]);
        if(count($present)>0){
            // map roll to student id
            $in = $pdo->prepare('SELECT id, roll FROM students WHERE roll = ? LIMIT 1');
            $ins = $pdo->prepare('INSERT INTO attendance (student_id, date) VALUES (?,?)');
            foreach($present as $roll){
                $in->execute([$roll]);
                $row = $in->fetch(PDO::FETCH_ASSOC);
                if($row){
                    $ins->execute([$row['id'],$date]);
                }
            }
        }
        echo json_encode(['ok'=>true]);
        exit;
    } elseif($action === 'get_attendance'){
        $date = $_GET['date'];
        $stmt = $pdo->prepare('SELECT s.roll, s.name FROM attendance a JOIN students s ON a.student_id =
        $stmt->execute([$date]);
        $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
        echo json_encode(['present'=>$rows]);
        exit;
    }
    echo json_encode(['error'=>'no action']);
```

```
} catch(Exception $e){
    http_response_code(500);
    echo json_encode(['error'=>$e->getMessage()]);
}
?>
```

5) README / Instructions

```
Instructions to run:
1. Place index.html, styles.css, script.js, and api.php in the same folder on a PHP-enabled server (PHP 7
2. Ensure the folder is writable by the web server (api.php will create attendance.db).
3. Open index.html in the browser (via the server URL, e.g. http://localhost/attendance/index.html).
4. Add students, choose a date and mark attendance, then Save. Use the Report section to view attendance

Notes & improvements you can add:
- Add authentication (login) for teachers.
- Add export to CSV feature.
- Use MySQL if you prefer a centralized DB for multi-user setups.
- Add pagination and search for large student lists.
- Add UI improvements (Bootstrap / Tailwind) and validations.
```