Task-1

Description: Build a student grade tracker using advanced JavaScript and ES6 features for data management.

Requirements:

1. Create a grade management system demonstrating ES6 features and advanced JavaScript concepts.
2. ES6 Features Implementation (Must Use All):
   - Arrow Functions: 4 different arrow functions for calculations and data processing
   - Let/Const: Replace all var declarations with appropriate let/const usage
   - Template Literals: 3 template literals for dynamic string creation and formatting
   - Destructuring: Array and object destructuring for student data extraction
   - Spread Operator: Use spread for array copying and object merging
3. Advanced JavaScript Concepts:
   - Array Methods: Use map(), filter(), reduce(), find() for grade calculations
   - Higher-Order Functions: Functions that accept other functions as parameters
   - Promises: Basic promise implementation for simulated data loading
   - Modules: Separate utility functions into modules (if using modern environment)

Application features:

- Add student records with multiple subject grades
- Calculate average grades using reduce() method
- Filter students by grade ranges using filter()
- Search students using find() method
- Display formatted grade reports using template literals
- Sort students by different criteria (name, grade, etc.)

Technical implementation:

- Student data stored in array of objects
- Grade calculation functions using ES6 arrow syntax
- Dynamic UI updates using template literals
- Form validation with ES6 features
- Local data persistence simulation

Deliverables:

- HTML interface for grade management
- JavaScript file showcasing all required ES6 features
- Documentation of ES6 features used with code examples
- Test data with 10 student records
- Feature demonstration screenshots

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>Data Management | Information System</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Outfit:wght@300;400;500;600;700&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <div class="app-container">
    <!-- Sidebar Navigation -->
    <aside class="sidebar">
      <div class="logo-section">
        <div class="logo-icon">DS</div>
        <h1>Data System</h1>
      </div>
      <nav class="nav-menu">
        <a href="#" class="nav-item active" data-view="dashboard">
          <span class="icon"></span> Overview
        </a>
        <a href="#" class="nav-item" data-view="students">
          <span class="icon"></span> Entries
        </a>
        <a href="#" class="nav-item" data-view="add-student">
          <span class="icon"></span> Add Entry
        </a>
        <a href="#" class="nav-item" data-view="reports">
          <span class="icon"></span> Summary
        </a>
      </nav>
      <div class="sidebar-footer">
        <p>© 2026 Data Management System</p>
      </div>
    </aside>

    <!-- Main Content Area -->
    <main class="main-content">
      <!-- Header -->
      <header class="top-header">
        <div class="search-bar">
          <input type="text" id="globalSearch" placeholder="Search entry by name...">
```

```
    </div>
    <div class="user-profile">
      <span id="currentTime"></span>
      <div class="avatar">User</div>
    </div>
  </header>

  <!-- Dynamic Views -->
  <div id="view-container">
    <!-- Dashboard View -->
    <section id="dashboard-view" class="view">
      <div class="stats-grid">
        <div class="stat-card">
          <h3>Total Entries</h3>
          <p id="totalStudentsCount">0</p>
        </div>
        <div class="stat-card grade-a">
          <h3>System Average</h3>
          <p id="classAverageScore">0%</p>
        </div>
        <div class="stat-card">
          <h3>Highest Entry</h3>
          <p id="topPerformerName">-</p>
        </div>
        <div class="stat-card">
          <h3>Passing Rate</h3>
          <p id="passingRate">0%</p>
        </div>
      </div>

      <div class="recent-students">
        <h2>Data Range Overview</h2>
        <div class="table-wrapper">
          <table id="dashboardTable">
            <thead>
              <tr>
                <th>Name</th>
                <th>Value 1</th>
                <th>Value 2</th>
                <th>Value 3</th>
                <th>Average</th>
                <th>Level</th>
              </tr>
            </thead>
            <tbody id="dashboardList">
              <!-- To be populated -->
            </tbody>
          </table>
        </div>
      </div>
    </section>
```

```
<!-- Students List View -->
<section id="students-view" class="view hidden">
  <div class="view-header">
    <h2>Data View</h2>
    <div class="filter-controls">
      <select id="gradeFilter">
        <option value="all">All Groups</option>
        <option value="A">Group A (>= 90)</option>
        <option value="B">Group B (>= 80)</option>
        <option value="C">Group C (>= 70)</option>
        <option value="D">Group D (>= 60)</option>
        <option value="F">Group F (< 60)</option>
      </select>
      <select id="sortControl">
        <option value="name-asc">Name (A-Z)</option>
        <option value="name-desc">Name (Z-A)</option>
        <option value="grade-desc">Level (High-Low)</option>
        <option value="grade-asc">Level (Low-High)</option>
      </select>
    </div>
  </div>
  <div class="student-cards-grid" id="studentCardsContainer">
    <!-- Student cards will be injected here -->
  </div>
</section>

<!-- Add Student View -->
<section id="add-student-view" class="view hidden">
  <div class="form-container">
    <h2>Create New Entry</h2>
    <form id="studentForm">
      <div class="form-group">
        <label for="studentName">Full Name</label>
        <input type="text" id="studentName" required placeholder="e.g. Person 11">
      </div>
      <div class="form-group">
        <label for="studentId">Student ID</label>
        <input type="text" id="studentId" required placeholder="e.g. ID-1024">
      </div>
      <div class="grades-input-grid">
        <div class="form-group">
          <label for="mathGrade">Value 1</label>
          <input type="number" id="mathGrade" min="0" max="100" required>
        </div>
        <div class="form-group">
          <label for="scienceGrade">Value 2</label>
          <input type="number" id="scienceGrade" min="0" max="100" required>
        </div>
        <div class="form-group">
          <label for="englishGrade">Value 3</label>
```

```
                    <input type="number" id="englishGrade" min="0" max="100" required>
                  </div>
                  <div class="form-group">
                    <label for="historyGrade">Value 4</label>
                    <input type="number" id="historyGrade" min="0" max="100" required>
                  </div>
                </div>
                <button type="submit" class="btn-primary">Add Entry</button>
              </form>
            </div>
          </section>

          <!-- Reports View -->
          <section id="reports-view" class="view hidden">
            <div class="report-header">
              <h2>Data Summary</h2>
              <button class="btn-secondary" id="downloadReport">Download Data</button>
            </div>
            <div id="reportSummary" class="report-summary-box">
              <!-- Summary generated by template literals -->
            </div>
            <div class="chart-simulation" id="gradeChart">
              <!-- Visual representation simulation -->
            </div>
          </section>
        </div>
      </main>
    </div>

    <!-- Loading Overlay -->
    <div id="loadingOverlay" class="loading-overlay hidden">
      <div class="spinner"></div>
      <p>Processing...</p>
    </div>

    <!-- Notification system -->
    <div id="notificationContainer"></div>

    <script type="module" src="js/app.js"></script>
</body>

</html>
```

style.css:

```
:root {
    --primary: #6366f1;
    --primary-hover: #4f46e5;
    --bg-main: #f8fafc;
    --bg-white: #ffffff;
    --sidebar-bg: #0f172a;
    --text-main: #1e293b;
```

```
    --text-muted: #64748b;
    --accent-success: #10b981;
    --accent-warning: #f59e0b;
    --accent-danger: #ef4444;
    --radius-lg: 12px;
    --radius-md: 8px;
    --shadow: 0 4px 6px -1px rgb(0 0 0 / 0.1), 0 2px 4px -2px rgb(0 0 0 / 0.1);
    --transition: all 0.3s ease;
}

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: 'Outfit', sans-serif;
    background-color: var(--bg-main);
    color: var(--text-main);
    line-height: 1.6;
    overflow-x: hidden;
}

/* Layout */
.app-container {
    display: flex;
    min-height: 100vh;
}

/* Sidebar */
.sidebar {
    width: 260px;
    background-color: var(--sidebar-bg);
    color: white;
    padding: 2rem 1.5rem;
    display: flex;
    flex-direction: column;
    position: fixed;
    height: 100vh;
    z-index: 100;
}

.logo-section {
    display: flex;
    align-items: center;
    gap: 0.75rem;
    margin-bottom: 3rem;
}

.logo-icon {
```

```
    background: var(--primary);
    color: white;
    width: 40px;
    height: 40px;
    display: flex;
    align-items: center;
    justify-content: center;
    border-radius: 10px;
    font-weight: 700;
    font-size: 1.2rem;
}

.logo-section h1 {
    font-size: 1.25rem;
    font-weight: 600;
}

.nav-menu {
    flex-grow: 1;
}

.nav-item {
    display: flex;
    align-items: center;
    gap: 1rem;
    padding: 0.75rem 1rem;
    color: #94a3b8;
    text-decoration: none;
    border-radius: var(--radius-md);
    margin-bottom: 0.5rem;
    transition: var(--transition);
}

.nav-item:hover,
.nav-item.active {
    background-color: rgba(255, 255, 255, 0.1);
    color: white;
}

.nav-item.active {
    background-color: var(--primary);
}

.sidebar-footer {
    font-size: 0.8rem;
    color: #64748b;
    text-align: center;
}

/* Main Content */
.main-content {
```

```
    flex-grow: 1;
    margin-left: 260px;
    padding: 2rem;
    min-width: 0;
}

.top-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 2rem;
}

.search-bar input {
    padding: 0.75rem 1.5rem;
    border: 1px solid #e2e8f0;
    border-radius: 30px;
    width: 350px;
    font-family: inherit;
    outline: none;
    transition: var(--transition);
}

.search-bar input:focus {
    border-color: var(--primary);
    box-shadow: 0 0 0 4px rgba(99, 102, 241, 0.1);
}

.user-profile {
    display: flex;
    align-items: center;
    gap: 1.5rem;
}

.avatar {
    width: 40px;
    height: 40px;
    background: #e2e8f0;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: 600;
    font-size: 0.8rem;
}

/* Dashboard Stats */
.stats-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 1.5rem;
```

```
    margin-bottom: 2rem;
}

.stat-card {
    background: var(--bg-white);
    padding: 1.5rem;
    border-radius: var(--radius-lg);
    box-shadow: var(--shadow);
    transition: var(--transition);
}

.stat-card:hover {
    transform: translateY(-5px);
}

.stat-card h3 {
    font-size: 0.875rem;
    color: var(--text-muted);
    margin-bottom: 0.5rem;
}

.stat-card p {
    font-size: 1.875rem;
    font-weight: 700;
    color: var(--text-main);
}

/* Views */
.view {
    animation: fadeIn 0.4s ease-out;
}

@keyframes fadeIn {
    from {
        opacity: 0;
        transform: translateY(10px);
    }

    to {
        opacity: 1;
        transform: translateY(0);
    }
}

/* Tables */
.table-wrapper {
    background: var(--bg-white);
    border-radius: var(--radius-lg);
    box-shadow: var(--shadow);
    overflow: hidden;
    margin-top: 1rem;
```

```
}

table {
  width: 100%;
  border-collapse: collapse;
}

th {
  background: #f1f5f9;
  padding: 1rem;
  text-align: left;
  font-size: 0.8rem;
  text-transform: uppercase;
  color: var(--text-muted);
  font-weight: 600;
}

td {
  padding: 1rem;
  border-bottom: 1px solid #f1f5f9;
}

.grade-pill {
  padding: 0.25rem 0.75rem;
  border-radius: 20px;
  font-size: 0.75rem;
  font-weight: 600;
}

.grade-A {
  background: #dcfce7;
  color: #166534;
}

.grade-B {
  background: #dbeafe;
  color: #1e40af;
}

.grade-C {
  background: #fef9c3;
  color: #854d0e;
}

.grade-D {
  background: #ffedd5;
  color: #9a3412;
}

.grade-F {
  background: #fee2e2;
```

```
    color: #991b1b;
}

/* Student Cards (Students View) */
.student-cards-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 1.5rem;
}

.student-card {
    background: var(--bg-white);
    border-radius: var(--radius-lg);
    padding: 1.5rem;
    box-shadow: var(--shadow);
    display: flex;
    flex-direction: column;
    gap: 1rem;
}

.card-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.student-info h4 {
    font-size: 1.1rem;
    margin-bottom: 0.25rem;
}

.student-info span {
    color: var(--text-muted);
    font-size: 0.8rem;
}

.subjects-row {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 0.75rem;
}

.subject-item {
    font-size: 0.875rem;
    display: flex;
    justify-content: space-between;
    padding: 0.5rem;
    background: #f8fafc;
    border-radius: var(--radius-md);
}
```

```
/* Form Styling */
.form-container {
  max-width: 600px;
  margin: 0 auto;
  background: var(--bg-white);
  padding: 2.5rem;
  border-radius: var(--radius-lg);
  box-shadow: var(--shadow);
}

.form-container h2 {
  margin-bottom: 2rem;
}

.form-group {
  margin-bottom: 1.5rem;
}

.form-group label {
  display: block;
  margin-bottom: 0.5rem;
  font-weight: 500;
  color: var(--text-main);
}

.form-group input {
  width: 100%;
  padding: 0.75rem 1rem;
  border: 1px solid #e2e8f0;
  border-radius: var(--radius-md);
  font-family: inherit;
}

.grades-input-grid {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 1rem;
}

.btn-primary {
  background: var(--primary);
  color: white;
  border: none;
  padding: 1rem 2rem;
  border-radius: var(--radius-md);
  font-weight: 600;
  cursor: pointer;
  width: 100%;
  font-family: inherit;
  transition: var(--transition);
}
```

```
.btn-primary:hover {
    background: var(--primary-hover);
}

/* View Controls */
.view-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 2rem;
}

.filter-controls {
    display: flex;
    gap: 1rem;
}

.filter-controls select {
    padding: 0.5rem 1rem;
    border-radius: var(--radius-md);
    border: 1px solid #e2e8f0;
    font-family: inherit;
    outline: none;
}

/* Loading Overlay */
.loading-overlay {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(255, 255, 255, 0.9);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    z-index: 1000;
}

.spinner {
    width: 40px;
    height: 40px;
    border: 4px solid #f3f3f3;
    border-top: 4px solid var(--primary);
    border-radius: 50%;
    animation: spin 1s linear infinite;
    margin-bottom: 1rem;
}
```

```
@keyframes spin {
  0% {
    transform: rotate(0deg);
  }

  100% {
    transform: rotate(360deg);
  }
}

/* Report Styles */
.report-summary-box {
  background: #eff6ff;
  border-left: 4px solid var(--primary);
  padding: 1.5rem;
  border-radius: var(--radius-md);
  margin-top: 1rem;
}

/* Responsive */
@media (max-width: 768px) {
  .sidebar {
    transform: translateX(-100%);
    width: 100%;
    height: auto;
    position: relative;
  }

  .sidebar.mobile-open {
    transform: translateX(0);
  }

  .main-content {
    margin-left: 0;
  }

  .stats-grid {
    grid-template-columns: 1fr;
  }

  .grades-input-grid {
    grid-template-columns: 1fr;
  }
}

.hidden {
  display: none !important;
}
```

js/app.js:

```
/**
```

```
 * Main Application Logic
 * Showcases: ES6 features, Spread operator, Object destructuring, Event handling
 */
import { initialStudents } from './data.js';
import {
    calculateAverage,
    getLetterGrade,
    filterByGrade,
    findStudent,
    storage,
    simulateAsyncLoad
} from './utils.js';

// Application State
let students = [];
let currentView = 'dashboard';

// DOM Elements
const viewContainer = document.getElementById('view-container');
const studentForm = document.getElementById('studentForm');
const dashboardList = document.getElementById('dashboardList');
const studentCardsContainer = document.getElementById('studentCardsContainer');
const loadingOverlay = document.getElementById('loadingOverlay');

/**
 * Initialize Application
 */
const init = async () => {
    loadingOverlay.classList.remove('hidden');

    // Load from storage or use initial data
    const savedData = storage.load('gm_students');
    const rawData = savedData || initialStudents;

    // Demonstrate Promise usage
    students = await simulateAsyncLoad(rawData);

    loadingOverlay.classList.add('hidden');
    updateUI();
    setupEventListeners();
};

/**
 * Update the whole UI based on current state
 */
const updateUI = () => {
    renderDashboard();
    renderStudentList();
    renderStats();
    renderReport();
};
```

```
/**
 * Render Dashboard Table
 * Showcases: map(), Template literals
 */
const renderDashboard = () => {
  dashboardList.innerHTML = students.slice(0, 5).map(student => {
    const { name, grades } = student; // Object destructuring
    const avg = calculateAverage(grades);
    const letter = getLetterGrade(avg);

    return `
      <tr>
        <td><strong>${name}</strong></td>
        <td>${grades.value1}</td>
        <td>${grades.value2}</td>
        <td>${grades.value3}</td>
        <td>${avg}%</td>
        <td><span class="grade-pill grade-${letter}">${letter}</span></td>
      </tr>
    `;
  }).join('');
};

/**
 * Render Full Student List with Filters
 * Showcases: filter(), map()
 */
const renderStudentList = () => {
  const filterVal = document.getElementById('gradeFilter').value;
  const sortVal = document.getElementById('sortControl').value;

  let filtered = filterByGrade(students, filterVal);

  // Sort logic
  filtered.sort((a, b) => {
    if (sortVal === 'name-asc') return a.name.localeCompare(b.name);
    if (sortVal === 'name-desc') return b.name.localeCompare(a.name);
    const avgA = parseFloat(calculateAverage(a.grades));
    const avgB = parseFloat(calculateAverage(b.grades));
    return sortVal === 'grade-desc' ? avgB - avgA : avgA - avgB;
  });

  studentCardsContainer.innerHTML = filtered.map(student => {
    const { name, id, grades } = student;
    const avg = calculateAverage(grades);
    const letter = getLetterGrade(avg);

    return `
      <div class="student-card">
        <div class="card-header">
```

```
            <div class="student-info">
              <h4>${name}</h4>
              <span>ID: ${id}</span>
            </div>
            <span class="grade-pill grade-${letter}">${letter}</span>
          </div>
          <div class="subjects-row">
            ${Object.entries(grades).map(([subj, score]) => `
              <div class="subject-item">
                <span style="text-transform: capitalize;">${subj}</span>
                <strong>${score}</strong>
              </div>
            `).join('')}
          </div>
          <div class="card-footer" style="margin-top: auto; display: flex; justify-content: space-
between; align-items: center;">
            <span style="font-size: 0.9rem; font-weight: 600;">AVG: ${avg}%</span>
            <button class="btn-delete" data-id="${id}" style="color: var(--accent-danger); border:
none; background: none; cursor: pointer; font-size: 0.8rem;">Delete</button>
          </div>
        </div>
      `;
  }).join('');
};

/**
 * Render Statistics View
 * Showcases: reduce(), arrow functions
 */
const renderStats = () => {
  const total = students.length;
  document.getElementById('totalStudentsCount').textContent = total;

  if (total === 0) return;

  // Calculate system average using reduce
  const systemAvg = (students.reduce((acc, s) => acc + parseFloat(calculateAverage(s.grades)), 0) /
total).toFixed(1);
  document.getElementById('classAverageScore').textContent = `${systemAvg}%`;

  // Find top performer using reduce
  const topPerformer = students.reduce((top, s) => {
    const curAvg = parseFloat(calculateAverage(s.grades));
    const topAvg = parseFloat(calculateAverage(top.grades));
    return curAvg > topAvg ? s : top;
  });
  document.getElementById('topPerformerName').textContent = topPerformer.name;

  // Calculate passing rate (avg >= 60)
  const passingCount = students.filter(s => calculateAverage(s.grades) >= 60).length;
  document.getElementById('passingRate').textContent = `${((passingCount / total) *
```

```
100).toFixed(0)}%`;
};

/**
 * Render Reports View
 */
const renderReport = () => {
   const reportSummary = document.getElementById('reportSummary');
   const sortedStudents = [...students].sort((a,b) => calculateAverage(b.grades) -
calculateAverage(a.grades));

   const top3 = sortedStudents.slice(0, 3);
   const bottom3 = sortedStudents.slice(-3).reverse();

   // Showcase template literals for complex strings
   reportSummary.innerHTML = `
      <div class="report-section">
         <h3 style="margin-bottom: 1rem;">Highest Entries</h3>
         <ul>
            ${top3.map(s => `<li>${s.name} - ${calculateAverage(s.grades)}%</li>`).join('')}
         </ul>
      </div>
      <hr style="margin: 1.5rem 0; border: 0; border-top: 1px solid #e2e8f0;">
      <div class="report-section">
         <h3 style="margin-bottom: 1rem;">Entries Below Range</h3>
         <ul>
            ${bottom3.map(s => `<li>${s.name} - ${calculateAverage(s.grades)}%</li>`).join('')}
         </ul>
      </div>
   `;
};

/**
 * Event Listeners
 */
const setupEventListeners = () => {
   // Navigation
   document.querySelectorAll('.nav-item').forEach(item => {
      item.addEventListener('click', (e) => {
         e.preventDefault();
         const view = item.getAttribute('data-view');
         switchView(view);
      });
   });

   // Form Submission
   studentForm.addEventListener('submit', (e) => {
      e.preventDefault();

      // Destructure input values
      const name = document.getElementById('studentName').value.trim();
```

```
    const id = document.getElementById('studentId').value.trim();

    // Custom ES6 Validation using some() and every()
    const gradesInputs = ['mathGrade', 'scienceGrade', 'englishGrade', 'historyGrade'];
    const isGradesValid = gradesInputs.every(id => {
        const val = parseInt(document.getElementById(id).value);
        return !isNaN(val) && val >= 0 && val <= 100;
    });

    if (name.length < 3 || !isGradesValid) {
        showNotification("Please ensure all fields are valid.", "error");
        return;
    }

    const grades = {
        value1: parseInt(document.getElementById('mathGrade').value),
        value2: parseInt(document.getElementById('scienceGrade').value),
        value3: parseInt(document.getElementById('englishGrade').value),
        value4: parseInt(document.getElementById('historyGrade').value)
    };

    // Create new record using spread for merging if needed
    const newStudent = {
        id,
        name,
        grades,
        addedAt: new Date().toISOString()
    };

    // Update state using spread operator
    students = [newStudent, ...students];

    storage.save('gm_students', students);
    showNotification(`Entry ${name} added.`);
    studentForm.reset();
    switchView('dashboard');
    updateUI();
});

// Search
document.getElementById('globalSearch').addEventListener('input', (e) => {
    const term = e.target.value.toLowerCase();
    if (term.length > 2) {
        const found = findStudent(students, term);
        if (found) {
            // Focus or highlight search behavior
            console.log('Found:', found.name);
        }
    }
    // Actually we might want search to filter the current view
    // For simplicity, search will trigger filter in students view
```

```
    if (currentView === 'students') renderStudentList();
});

// Filters
document.getElementById('gradeFilter').addEventListener('change', renderStudentList);
document.getElementById('sortControl').addEventListener('change', renderStudentList);

// Dynamic deletions
studentCardsContainer.addEventListener('click', (e) => {
    if (e.target.classList.contains('btn-delete')) {
        const id = e.target.getAttribute('data-id');
        students = students.filter(s => s.id !== id);
        storage.save('gm_students', students);
        updateUI();
    }
});

// Time display
setInterval(() => {
    document.getElementById('currentTime').textContent = new Date().toLocaleTimeString();
}, 1000);

// CSV Download
document.getElementById('downloadReport').addEventListener('click', () => {
    const headers = ['ID', 'Name', 'Value 1', 'Value 2', 'Value 3', 'Value 4', 'Average', 'Level'];
    const csvData = students.map(s => {
        const avg = calculateAverage(s.grades);
        const letter = getLetterGrade(avg);
        return [
            s.id,
            s.name,
            s.grades.value1,
            s.grades.value2,
            s.grades.value3,
            s.grades.value4,
            avg,
            letter
        ].join(',');
    });

    const csvContent = [headers.join(','), ...csvData].join('\n');
    const blob = new Blob([csvContent], { type: 'text/csv' });
    const url = window.URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.setAttribute('hidden', '');
    a.setAttribute('href', url);
    a.setAttribute('download', 'data_export.csv');
    document.body.appendChild(a);
    a.click();
    document.body.removeChild(a);
    showNotification("Data exported to CSV.");
```

```
    });
};

const switchView = (viewName) => {
    currentView = viewName;
    document.querySelectorAll('.view').forEach(v => v.classList.add('hidden'));
    document.getElementById(`${viewName}-view`).classList.remove('hidden');

    document.querySelectorAll('.nav-item').forEach(nav => {
        nav.classList.toggle('active', nav.getAttribute('data-view') === viewName);
    });
};

const showNotification = (msg) => {
    const container = document.getElementById('notificationContainer');
    const toast = document.createElement('div');
    toast.className = 'toast';
    toast.style.cssText = `
        position: fixed; bottom: 20px; right: 20px;
        background: var(--sidebar-bg); color: white;
        padding: 1rem 2rem; border-radius: var(--radius-md);
        box-shadow: var(--shadow); z-index: 2000;
        animation: slideIn 0.3s ease-out;
    `;
    toast.textContent = msg;
    container.appendChild(toast);
    setTimeout(() => {
        toast.style.opacity = '0';
        setTimeout(() => toast.remove(), 300);
    }, 3000);
};

// Start the app
init();
```

js/data.js:

```
/**
 * initialStudents - Test Data with 10 records
 * Showcases: Array of objects, template literals for IDs
 * Updated with abstract names as requested (Person 1, Person 2, etc.)
 */
export const initialStudents = [
    {
        id: "ID-1001",
        name: "Person 1",
        grades: { value1: 92, value2: 88, value3: 95, value4: 90 },
        addedAt: new Date('2026-01-10').toISOString()
    },
    {
        id: "ID-1002",
        name: "Person 2",
```

```
    grades: { value1: 78, value2: 82, value3: 75, value4: 80 },
    addedAt: new Date('2026-01-12').toISOString()
  },
  {
    id: "ID-1003",
    name: "Person 3",
    grades: { value1: 95, value2: 98, value3: 92, value4: 96 },
    addedAt: new Date('2026-01-15').toISOString()
  },
  {
    id: "ID-1004",
    name: "Person 4",
    grades: { value1: 65, value2: 70, value3: 68, value4: 62 },
    addedAt: new Date('2026-01-18').toISOString()
  },
  {
    id: "ID-1005",
    name: "Person 5",
    grades: { value1: 88, value2: 85, value3: 90, value4: 87 },
    addedAt: new Date('2026-01-20').toISOString()
  },
  {
    id: "ID-1006",
    name: "Person 6",
    grades: { value1: 55, value2: 60, value3: 58, value4: 52 },
    addedAt: new Date('2026-01-22').toISOString()
  },
  {
    id: "ID-1007",
    name: "Person 7",
    grades: { value1: 100, value2: 99, value3: 98, value4: 100 },
    addedAt: new Date('2026-01-25').toISOString()
  },
  {
    id: "ID-1008",
    name: "Person 8",
    grades: { value1: 82, value2: 79, value3: 85, value4: 81 },
    addedAt: new Date('2026-01-28').toISOString()
  },
  {
    id: "ID-1009",
    name: "Person 9",
    grades: { value1: 74, value2: 76, value3: 72, value4: 78 },
    addedAt: new Date('2026-02-01').toISOString()
  },
  {
    id: "ID-1010",
    name: "Person 10",
    grades: { value1: 91, value2: 89, value3: 93, value4: 88 },
    addedAt: new Date('2026-02-05').toISOString()
  }
```

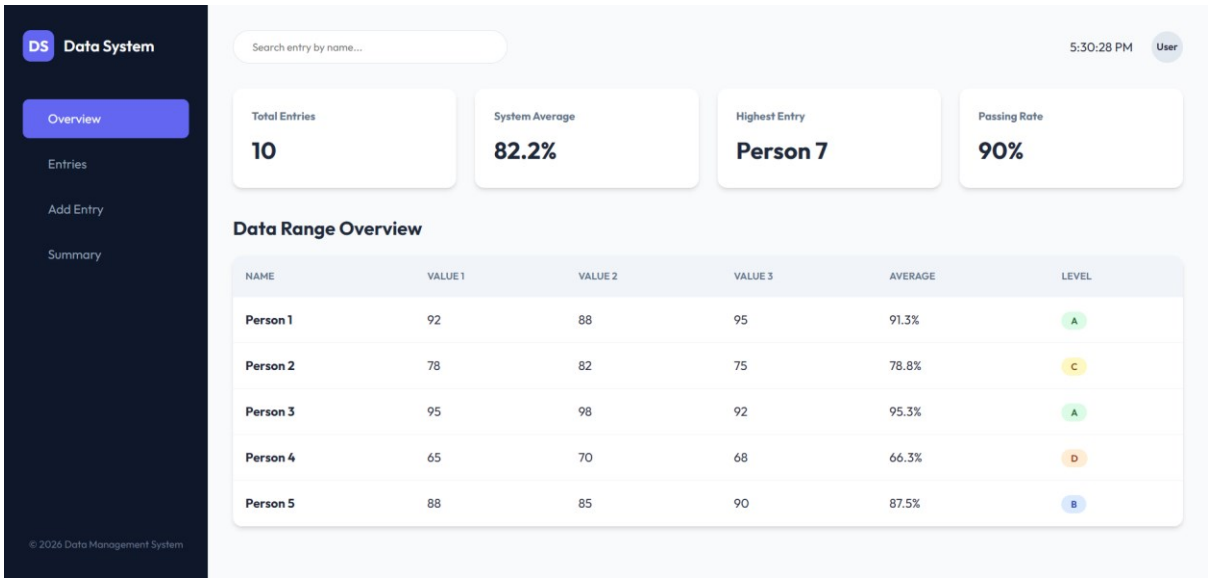
```
];
```

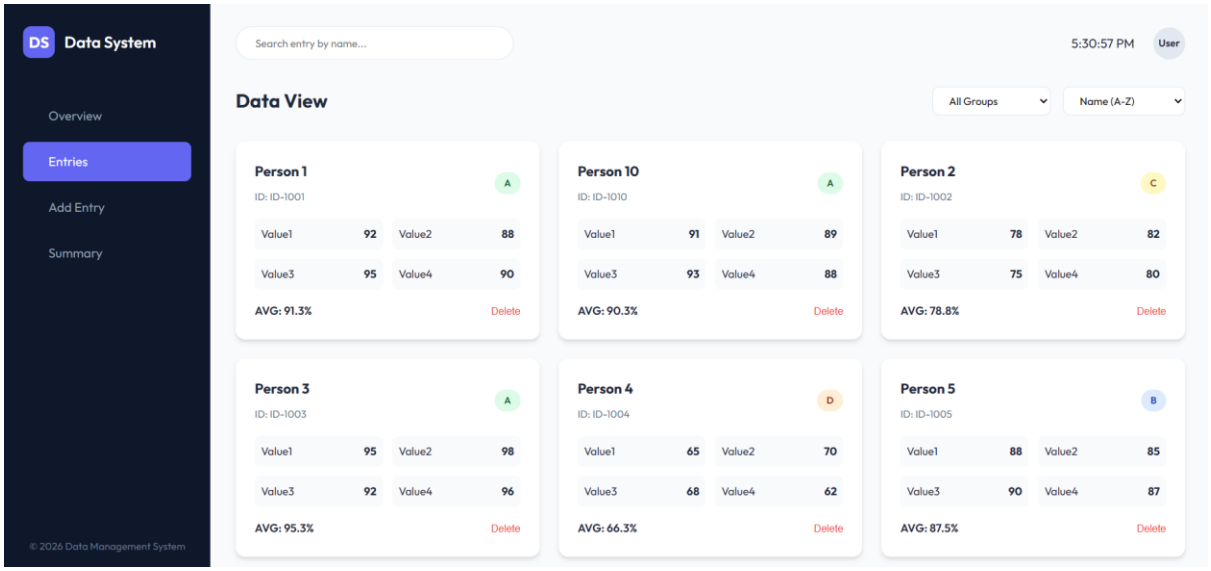
js/utils.js:

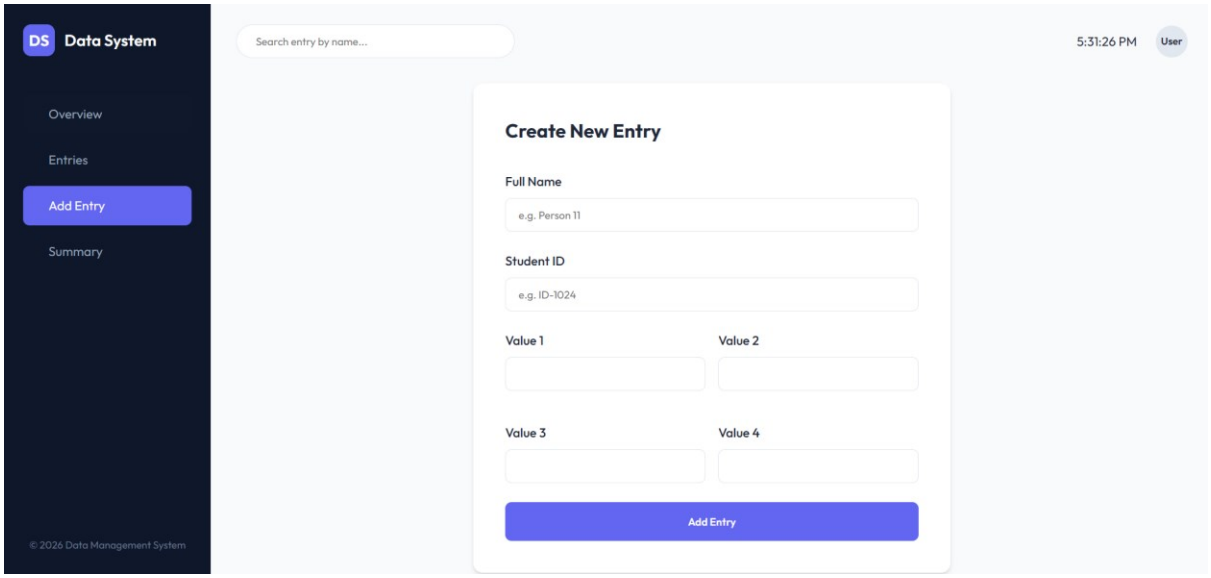
```
/**
 * Utility functions for Data Management
 * Showcases: ES6 features, Arrow functions, Array methods, Destructuring
 */

// 1. Arrow Function for Average Calculation (using reduce)
// Showcases: Arrow function, reduce()
export const calculateAverage = (grades) => {
   const values = Object.values(grades);
   const sum = values.reduce((acc, curr) => acc + curr, 0);
   return (sum / values.length).toFixed(1);
};

// 2. Arrow Function for Grade Mapping
// Showcases: Template literals, Arrow function
export const getLetterGrade = (avg) => {
   if (avg >= 90) return 'A';
   if (avg >= 80) return 'B';
   if (avg >= 70) return 'C';
   if (avg >= 60) return 'D';
   return 'F';
};

// 3. Higher-Order Function that accepts a calculation strategy
// Showcases: Higher-order function
export const processStudentData = (students, strategy) => {
   return students.map(student => strategy(student));
};

// 4. Formatter using Template Literals
// Showcases: Template literals, destructuring
export const formatStudentReport = ({ name, id, grades }) => {
   const avg = calculateAverage(grades);
   const grade = getLetterGrade(avg);
   return `
      <div class="report-card">
         <h4>${name} (${id})</h4>
         <p>Level: <strong>${grade}</strong></p>
         <p>Total: ${avg}%</p>
      </div>
   `;
};

// 5. Filtering students by grade range
// Showcases: filter(), Arrow function
export const filterByGrade = (students, targetGrade) => {
   if (targetGrade === 'all') return [...students]; // Spread operator for copying
   return students.filter(student => {
```

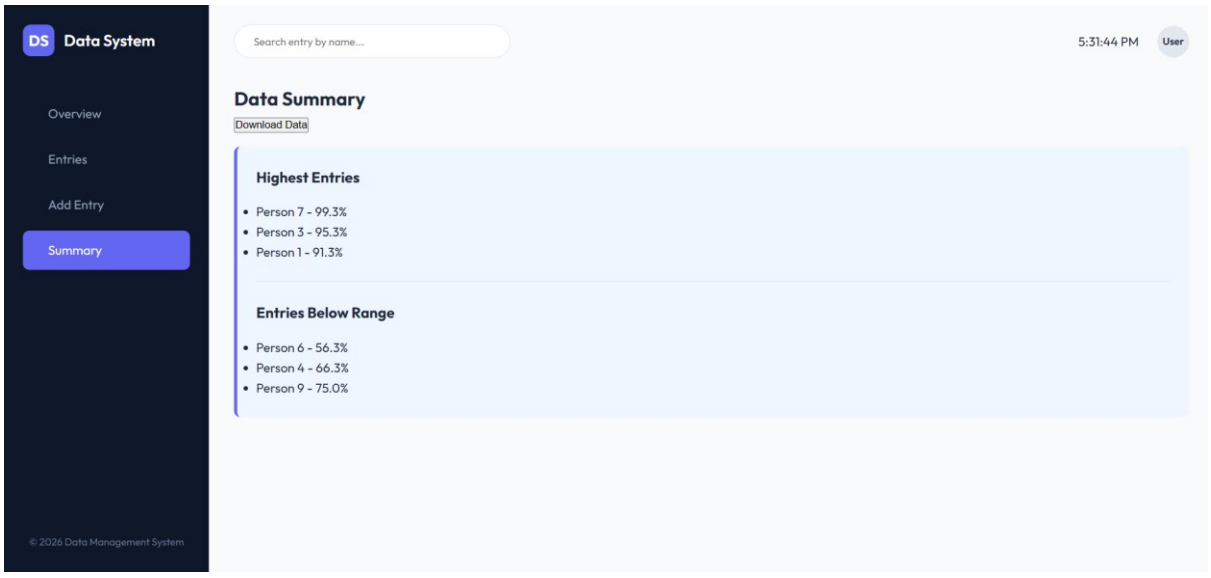
```
    const avg = calculateAverage(student.grades);
    return getLetterGrade(avg) === targetGrade;
  });
};

// 6. Searching student by name
// Showcases: find(), Arrow function
export const findStudent = (students, searchTerm) => {
  return students.find(s => s.name.toLowerCase().includes(searchTerm.toLowerCase()));
};

// 7. Simulating Data Persistence (Local Storage)
// Showcases: JSON stringify/parse
export const storage = {
  save: (key, data) => localStorage.setItem(key, JSON.stringify(data)),
  load: (key) => JSON.parse(localStorage.getItem(key)) || null
};

// 8. Promise implementation for simulated data loading
// Showcases: Promises
export const simulateAsyncLoad = (data) => {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve(data);
    }, 800);
  });
};
```

Output:




DS Data System
Overview
Entries
Add Entry
Summary
© 2026 Data Management System
Search entry by name...
5:30:57 PM
User
Data View
All Groups
Name (A-Z)
Person 1
ID: ID-1001
A
Value1 92
Value2 88
Value3 95
Value4 90
AVG: 91.3%
Delete
Person 10
ID: ID-1010
A
Value1 91
Value2 89
Value3 93
Value4 88
AVG: 90.3%
Delete
Person 2
ID: ID-1002
C
Value1 78
Value2 82
Value3 75
Value4 80
AVG: 78.8%
Delete
Person 3
ID: ID-1003
A
Value1 95
Value2 98
Value3 92
Value4 96
AVG: 95.3%
Delete
Person 4
ID: ID-1004
D
Value1 65
Value2 70
Value3 68
Value4 62
AVG: 66.3%
Delete
Person 5
ID: ID-1005
B
Value1 88
Value2 85
Value3 90
Value4 87
AVG: 87.5%
Delete


DS Data System
Overview
Entries
Add Entry
Summary
© 2026 Data Management System
Search entry by name...
5:31:26 PM
User
Create New Entry
Full Name
e.g. Person 11
Student ID
e.g. ID-1024
Value 1
Value 2
Value 3
Value 4
Add Entry


DS Data System
Overview
Entries
Add Entry
Summary
© 2026 Data Management System
Search entry by name...
5:31:44 PM
User
Data Summary
Download Data
Highest Entries
Person 7 - 99.3%
Person 3 - 95.3%
Person 1 - 91.3%
Entries Below Range
Person 6 - 56.3%
Person 4 - 66.3%
Person 9 - 75.0%