

# **Experiment 1.2**

**Student Name:** Subhradip Majumder

**Branch:** BE-AIT-CSE

**Semester:** 5th

**Subject Name:** Full Stack

**UID:** 23BAI70059

**Section/Group:** 23AML\_KRG-1

**Date of Performance:** 14 July 2025

**Subject Code:** 23CSP-339

1. **Problem Title:** To build an admin dashboard with dynamic theming using CSS Grid and JavaScript.
2. **Problem Description:**
   1. Create grid-based layout
   2. Implement dark/light mode toggle
   3. Design responsive sidebar
   4. Add data visualization placeholders
   5. Persist theme preference using localStorage
3. **Hardware Requirements:** 
   1. Processor: Intel i3/ Ryzen 3 or higher.
   2. RAM: 4GB minimum.
   3. Display: 1366x768 resolution.
4. **HTML:**

|  |
| --- |
| **<!DOCTYPE html>**  **<html lang="en">**  **<head>**  **<meta charset="UTF-8" />**  **<meta name="viewport" content="width=device-width, initial-scale=1.0"/>**  **<title>Admin Dashboard</title>**  **<link rel="stylesheet" href="style.css" />**  **</head>**  **<body>**  **<div class="dashboard">**  **<!-- Sidebar -->**  **<aside class="sidebar">**  **<h2>Admin</h2>**  **<nav>**  **<a href="#">Dashboard</a>**  **<a href="#">Users</a>**  **<a href="#">Analytics</a>**  **<a href="#">Settings</a>**  **</nav>**  **<button id="theme-toggle">Toggle Theme</button>**  **</aside>**  **<!-- Main Content -->**  **<main class="main">**  **<header class="header">**  **<h1>Welcome, Admin</h1>**  **</header>**  **<section class="cards">**  **<div class="card">Metric 1</div>**  **<div class="card">Metric 2</div>**  **<div class="card">Metric 3</div>**  **<div class="card">Metric 4</div>**  **</section>**  **<section class="charts">**  **<div class="chart-placeholder">Chart 1 Placeholder</div>**  **<div class="chart-placeholder">Chart 2 Placeholder</div>**  **</section>**  **</main>**    **</div>**  **<script src="script.js"></script>**  **</body>**  **</html>** |

1. **CSS:**

|  |
| --- |
| :root {  --bg-color: #f4f4f4;  --text-color: #333;  --card-bg: #fff;  --sidebar-bg: #2f3542;  --sidebar-text: #fff;  }  body.dark {  --bg-color: #1e1e1e;  --text-color: #f4f4f4;  --card-bg: #2c2c2c;  --sidebar-bg: #111;  --sidebar-text: #ccc;  }  \* {  box-sizing: border-box;  margin: 0;  padding: 0;  }  body {  font-family: 'Segoe UI', sans-serif;  background-color: var(--bg-color);  color: var(--text-color);  transition: background-color 0.3s ease;  }  .dashboard {  display: grid;  grid-template-columns: 250px 1fr;  min-height: 100vh;  }  .sidebar {  background-color: var(--sidebar-bg);  color: var(--sidebar-text);  padding: 20px;  }  .sidebar h2 {  margin-bottom: 20px;  }  .sidebar nav a {  display: block;  color: var(--sidebar-text);  text-decoration: none;  margin-bottom: 10px;  }  .sidebar button {  margin-top: 20px;  padding: 10px;  width: 100%;  cursor: pointer;  }  .main {  padding: 20px;  }  .header {  margin-bottom: 20px;  }  .cards {  display: grid;  grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));  gap: 20px;  margin-bottom: 30px;  }  .card {  background-color: var(--card-bg);  padding: 20px;  border-radius: 8px;  box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);  transition: background-color 0.3s ease;  }  .charts {  display: grid;  grid-template-columns: 1fr 1fr;  gap: 20px;  }  .chart-placeholder {  background-color: var(--card-bg);  height: 200px;  border-radius: 8px;  display: flex;  align-items: center;  justify-content: center;  font-weight: bold;  transition: background-color 0.3s ease;  }  @media (max-width: 768px) {  .dashboard {  grid-template-columns: 1fr;  }  .sidebar {  display: flex;  justify-content: space-between;  align-items: center;  }  .charts {  grid-template-columns: 1fr;  }  } |

1. **JavaScript:**

|  |
| --- |
| const toggleBtn = document.getElementById("theme-toggle");  document.addEventListener("DOMContentLoaded", () => {  const theme = localStorage.getItem("theme");  if (theme === "dark") {  document.body.classList.add("dark");  }  });  toggleBtn.addEventListener("click", () => {  document.body.classList.toggle("dark");  const currentTheme = document.body.classList.contains("dark") ? "dark" : "light";  localStorage.setItem("theme", currentTheme);  }); |

1. **Learning Outcome:** 
   1. I learnt how to link and connect the script.js and css file with the Html.
   2. I learnt how to work with html file.
   3. I learnt how to write js file and css.

HARD - LEVEL

1. **Problem Title:** Financial Forecast Matching with Fallback Strategy
2. **Problem Description:** You are a Data Engineer at FinSight Corp, a company that models Net Present Value (NPV) projections for investment decisions. Your system maintains two key datasets:

Year\_tbl: Actual recorded NPV’s of various financial instruments over different years:

ID: Unique Financial instrument identifier.

YEAR: Year of record

NPV: Net Present Value in that year

Queries\_tbl: A list of instrument-year pairs for which stakeholders are requesting NPV values:

ID: Financial instrument identifier

YEAR: Year of interest.

Find the NPV of each query from the Queries table. Return the output order by ID and Year in the sorted form. However, not all ID-YEAR combinations in the Queries table are present in the Year\_tbl. If an NPV is missing for a requested combination, assume it to be 0 to maintain a consistent financial report.

1. **SQL Commands:**
   1. Create the tables.

|  |
| --- |
| CREATE TABLE Year\_tbl (  ID INT,  YEAR INT,  NPV INT  );  -- Create Queries table (requested values)  CREATE TABLE Queries (  ID INT,  YEAR INT  ); |

* 1. Insert the values.

|  |
| --- |
| INSERT INTO Year\_tbl (ID, YEAR, NPV) VALUES  (1, 2018, 100),  (7, 2020, 30),  (13, 2019, 40),  (1, 2019, 113),  (2, 2008, 121),  (3, 2009, 12),  (11, 2020, 99),  (7, 2019, 0);  INSERT INTO Queries (ID, YEAR) VALUES  (1, 2019),  (2, 2008),  (3, 2009),  (7, 2018),  (7, 2019),  (7, 2020),  (13, 2019); |

* 1. Use a subquery to count the number of courses under each department.

|  |
| --- |
| select q.id, q.year, Isnull(y.NPV, 0) [NPV]  from Queries as q  left outer join Year\_tbl as y on q.id = y.id and q.YEAR = y.YEAR  order by q.id; |

1. **Output:**

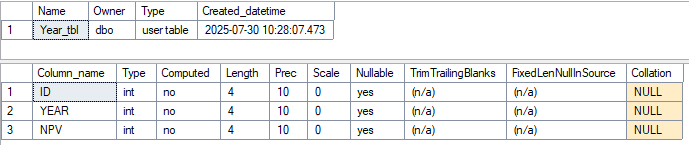


Figure 1 Year\_tbl description

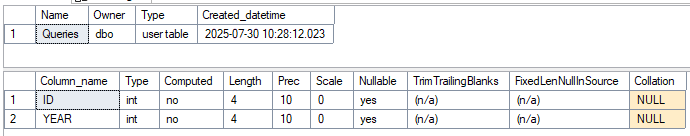


Figure 2 Queries table description

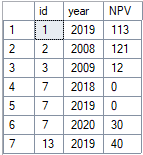


Figure 3 Select Query

1. **Learning Outcomes:**
   1. I learned how to perform left join and understand the table.
   2. I learned some of the build functions of the Microsoft SQL server.
   3. I learned about aliases in the SQL queries.