

## ปลดล็อกศักยภาพนักพัฒนา: ใช้ Gemini คู่กับ Google App Script เพื่อสร้าง Web App ได้เร็วขึ้น

ในโลกที่ความต้องการด้านดิจิทัลเติบโตอย่างรวดเร็ว การเรียนรู้ที่จะสร้างแอปพลิเคชันด้วยตัวเองไม่ใช่เรื่องไกลตัวอีกต่อไป แต่ก็ไม่ใช่เรื่องง่ายเสมอไปสำหรับทุกคน โชคดีที่เรามีเครื่องมืออัจฉริยะอย่าง Gemini เข้ามาเป็นผู้ช่วย Gemini คือโมเดล AI ที่มีความสามารถในการประมวลผลและสร้างสรรค์ข้อมูลในหลายรูปแบบ (multimodal) ไม่ว่าจะเป็นข้อความ รูปภาพ หรือโค้ด ทำให้ Gemini เป็นมากกว่าแค่ผู้ช่วย แต่เป็นเหมือน "คู่นักพัฒนา" ที่จะช่วยลดขั้นตอนที่ซับซ้อนและเร่งความเร็วในการสร้างโปรเจกต์ของคุณ บทความนี้จะแสดงให้เห็นว่า การใช้ Gemini ร่วมกับ Google App Script ซึ่งเป็นเครื่องมือสร้างแอปพลิเคชันบนคลาวด์ของ Google สามารถพลิกโฉมวิธีการทำงานของคุณให้ง่ายและรวดเร็วขึ้นได้อย่างไร

### ทำไม Gemini และ Google App Script ถึงเป็นคู่หูที่ลงตัว?

- **โค้ดที่เข้าใจภาษามนุษย์:** คุณไม่จำเป็นต้องเป็นผู้เชี่ยวชาญด้าน JavaScript เพื่อเริ่มเขียนโค้ด Google App Script เพียงแค่คุณอธิบายความต้องการของคุณเป็นภาษาไทยง่าย ๆ ให้ Gemini ฟัง เช่น "ฉันต้องการสร้างเว็บไซต์ที่มีปุ่ม 4 ปุ่มสำหรับแต่ละชีทใน Google Sheets" Gemini ก็สามารถสร้างโครงสร้างโค้ดพื้นฐานให้คุณได้ทันที
- **แก้ไขข้อผิดพลาด (Debugging) ได้อย่างรวดเร็ว:** เมื่อโค้ดของคุณเกิดข้อผิดพลาด (Error) และไม่สามารถทำงานได้ตามที่ตั้งใจไว้ คุณสามารถคัดลอกข้อความ Error นั้นแล้วนำไปให้ Gemini วิเคราะห์ได้เลย Gemini ไม่เพียงแต่จะบอกสาเหตุของปัญหา แต่ยังเสนอวิธีการแก้ไขพร้อมกับโค้ดที่ถูกต้องให้ด้วย ทำให้คุณประหยัดเวลาในการค้นหาต้นตอของปัญหาอย่างมาก
- **สร้างฟังก์ชันการทำงานที่ซับซ้อนได้ง่ายขึ้น:** ต้องการเพิ่มฟังก์ชันการค้นหา (Search Bar) หรือปุ่มกรองข้อมูล (Filter Buttons) แบบที่เห็นในตัวอย่างระบบตรวจสอบคะแนนใช่ไหม? ฟังก์ชันเหล่านี้มักจะต้องการโค้ดที่ซับซ้อน แต่ด้วยการอธิบายความต้องการอย่างชัดเจน Gemini สามารถสร้างโค้ด JavaScript ที่จัดการกับเหตุการณ์ (Event) ต่าง ๆ ในหน้าเว็บได้อย่างแม่นยำ

### ตัวอย่างการใช้งานจริง

เมื่อคุณต้องการสร้าง Web Application ระบบตรวจสอบคะแนน สิ่งที่คุณต้องทำคือ:

1. **บอกความต้องการ:** เริ่มต้นด้วยการให้คำสั่งที่ชัดเจนกับ Gemini เช่น "ฉันต้องการสร้าง web application ระบบตรวจสอบคะแนน ใน Google App Script โดยใช้ฐานข้อมูลจาก Google Sheets โครงสร้างของระบบต้องมี 4 ปุ่ม 'pretest', 'posttest', 'report', และ 'lms' ช่วยเขียนโค้ด GS, HTML และ JavaScript ให้หน่อย"
2. **รับโค้ดและปรับแต่ง:** Gemini จะให้โค้ดที่สมบูรณ์แบบครบทั้ง 3 ส่วนมาให้คุณ คุณเพียงแค่นำไปวางใน App Script Editor
3. **แก้ปัญหาเฉพาะหน้า:** หากโค้ดเกิด Error ในขั้นตอนการปรับใช้ (Deploy) คุณก็แค่คัดลอกข้อความผิดพลาดนั้นมาถาม Gemini เพื่อหาทางแก้ไข ซึ่งเป็นขั้นตอนที่สำคัญและประหยัดเวลาอย่างมาก

### เคล็ดลับในการใช้ Gemini ให้ได้ผลสูงสุด

- **ให้ข้อมูลที่ละเอียด:** ยิ่งคุณให้ข้อมูลกับ Gemini ได้ละเอียดมากเท่าไร โค้ดที่ได้ก็จะยิ่งตรงตามความต้องการของคุณมากขึ้นเท่านั้น เช่น ระบุชื่อชีท, ชื่อคอลัมน์, หรือรูปแบบหน้าต่างที่ต้องการ

- **แบ่งงานเป็นส่วนย่อย:** หากโปรเจกต์ของคุณใหญ่และซับซ้อน ให้แบ่งงานออกเป็นส่วนเล็ก ๆ เช่น เริ่มจากให้ Gemini สร้างหน้าตา (HTML) ก่อน จากนั้นจึงค่อยขอโค้ดสำหรับฟังก์ชันการดึงข้อมูล (GS) และฟังก์ชันการแสดงผล (JavaScript)
- **ถามคำถามปลายเปิด:** หากคุณไม่แน่ใจว่าควรจะใช้ฟังก์ชันไหน ลองถาม Gemini ว่า "ฉันควรจะใช้ฟังก์ชันอะไรเพื่อ..." หรือ "มีวิธีอื่นที่ดีกว่านี้ไหม?" ซึ่งจะช่วยให้คุณได้เรียนรู้และเห็นทางเลือกใหม่ ๆ

การใช้ Gemini ไม่ได้หมายความว่า คุณจะต้องหยุดเรียนรู้การเขียนโค้ด แต่เป็นการช่วย **เร่งกระบวนการเรียนรู้** และ **ปลดล็อกขีดจำกัด** ของคุณ ทำให้คุณสามารถเปลี่ยนไอเดียให้กลายเป็นจริงได้อย่างรวดเร็ว ไม่ว่าจะคุณจะเป็นนักพัฒนาที่กำลังมองหาเครื่องมือใหม่ ๆ หรือเป็นเพียงสมาชิกในชุมชนที่อยากทำให้งานของกลุ่มง่ายขึ้น Gemini และ Google App Script ก็เป็นคู่หูที่คุณไม่ควรพลาด

### ตัวอย่าง Prompt ที่ใช้สร้างระบบเช็คคะแนน

“ต้องการสร้าง web application ระบบตรวจสอบคะแนน ใน Google App Script โดยใช้ฐานข้อมูลที่มีอยู่ใน Google Sheets

โดยอยากให้ระบบมีหน้าตา interface ดังต่อไปนี้

- ให้นำหน้าตา interface ของ web application มีมี 4 ปุ่ม "pretest", "posttest", "report" และ "lms" เพื่อเข้าสู่ หน้า Dashboard แสดงคะแนน
- font ของระบบให้เป็น font ที่ชื่อว่า Kanit
- header title ของชื่อเว็บ ให้ background มีการออกแบบให้ดูสวยงาม ใช้โทนสี Soft tone อย่างน้อย 3 สี เช่น สีฟ้าอ่อน, สีเขียวมินต์, และสีชมพูอ่อน และ อิโมจิประกอบ
- ให้นำหน้าจอของ web application สามารถปรับเปลี่ยนขนาดแบบ responsive ได้กรณีแสดงผลลัพท์ในโทรศัพท์มือถือ

และอยากให้ระบบมี features ดังต่อไปนี้

- มี search bar ที่ userID สามารถพิมพ์ข้อความเพื่อ filter หาคะแนนที่ต้องการได้
  - มีปุ่มที่สามารถเลือกดูเฉพาะคะแนนแต่ละส่วนที่
- โครงสร้างไฟล์ Google Sheets ให้มีชีทดังต่อไปนี้
- ชีทคะแนนpretest สำหรับตรวจสอบคะแนน pretest
  - ชีทคะแนนposttest สำหรับตรวจสอบคะแนน posttest
  - ชีทคะแนนreport สำหรับตรวจสอบคะแนน repost
  - ชีทคะแนนlms สำหรับตรวจสอบคะแนน lms

spreadsheet

ID:

[https://docs.google.com/spreadsheets/d/1nAVtWQJ1n4EMs607TeVp6fFW3nzpP6Mv\\_U407lpNqT4/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1nAVtWQJ1n4EMs607TeVp6fFW3nzpP6Mv_U407lpNqT4/edit?usp=sharing)

ใช้ sweet alert แสดงสถานะขณะรอการบันทึกและบันทึกสำเร็จ เมื่อสร้างระบบเสร็จแล้วให้แสดงโค้ด GS (Google App Script) จนจบครบทุกบรรทัด ในหน้าแชทนี้ด้วย

\*\*\* เมื่อเปลี่ยนข้อความใน Google Sheet แล้วให้โปรแกรมแสดง และบันทึกคะแนนนักเรียน เปลี่ยนตามด้วย เช่น lab1,คะแนนรวม”

### ขั้นตอนการทำงานกับ Gemini

Gemini

2.5 Flash

เขียน prompt

Gemini จะสร้าง code ให้พร้อมทั้งอธิบาย  
การใช้งาน

ต้องการสร้าง web application ระบบตรวจสอบคะแนน ใน Google App Script โดยใช้ฐานข้อมูลที่มีอยู่ใน Google Sheets

- โดยอยากให้ระบบมีหน้าตา interface ดังต่อไปนี้
- ให้หน้าตา interface ของ web application มีมี 4 ปุ่ม "pretest", "posttest", "report" และ "lms" เพื่อเข้าสู่ หน้า Dashboard แสดงคะแนน
- font ของระบบให้เป็น font ที่ชื่อว่า Kanit
- header title ของชื่อเว็บ ให้ background มีการออกแบบให้ดูสวยงาม ใช้โทนสี Soft tone อย่างน้อย 3 สี เช่น สีฟ้าอ่อน, สีเขียว มินต์, และสีชมพูอ่อน และ อีโมจิประกอบ
- ให้หน้าจอของ web application สามารถปรับเปลี่ยนขนาดแบบ responsive ได้กรณีแสดงผลทั้งในโทรศัพท์มือถือ

### โครงสร้างไฟล์ใน Google App Script

คุณจะต้องสร้างไฟล์ 3 ไฟล์ในโปรเจกต์ Google App Script ของคุณ:

1. **Code.gs:** ไฟล์หลักสำหรับการจัดการกับคำขอจากฝั่ง Web App และดึงข้อมูลจาก Google Sheets
2. **Index.html:** ไฟล์ HTML หลักที่ประกอบด้วยโครงสร้างหน้าเว็บ ปุ่มต่างๆ และพื้นที่แสดงผล
3. **JavaScript.html:** ไฟล์ JavaScript สำหรับจัดการการทำงานฝั่ง Client-side

### โค้ด Google App Script (GS)

#### 1. Code.gs

JavaScript

```
/**
 * @file Code.gs
 * @description Google App Script code for handling web requests and
interacting with Google Sheets.
 */

// Define the ID of your Google Spreadsheet
const SPREADSHEET_ID = '1nAVtWQJ1n4EMs607TeVp6fFW3nzpP6Mv_U4071pNqT4';

/**
 * Serves the web application HTML page.
 * This function is automatically called when the web app is accessed.
 * @param {GoogleAppsScript.Events.DoGet} e - Event object for the GET
request.
 * @returns {GoogleAppsScript.HTML.HtmlOutput} The HTML page to be served.
```

```

*/
function doGet(e) {
  // Create an HTML output from the 'Index.html' file.
  // Set the title of the web page.
  // Add an X-UA-Compatible header for better browser compatibility.
  return HtmlService.createTemplateFromFile('Index')
    .evaluate()
    .setTitle('ระบบตรวจสอบคะแนน')
    .addMetaTag('viewport', 'width=device-width, initial-scale=1.0')
    .addMetaTag('charset', 'UTF-8')
    .setXFrameOptions(HtmlService.XFrameOptions.ALLOWALL); // Allow embedding
  in iframes if needed
}

/**
 * Includes the content of a specified HTML file into another HTML file.
 * This is a common pattern to modularize HTML, CSS, and JavaScript within
  GAS.
 * @param {string} filename - The name of the HTML file to include (without
  .html extension).
 * @returns {string} The content of the specified HTML file.
 */
function include(filename) {
  return HtmlService.createHtmlOutputFromFile(filename).getContent();
}

/**
 * Fetches all data from a specified sheet in the Google Spreadsheet.
 * The first row is treated as headers, and subsequent rows are data.
 * @param {string} sheetName - The name of the sheet to retrieve data from.
 * @returns {Array<Object>} An array of objects, where each object represents
  a row
 * and keys are column headers. Returns an empty array if sheet not found or
  no data.
 */
function getData(sheetName) {
  try {
    const spreadsheet = SpreadsheetApp.openById(SPREADSHEET_ID);
    const sheet = spreadsheet.getSheetByName(sheetName);

    if (!sheet) {
      console.error(`Sheet not found: ${sheetName}`);
      return []; // Return empty array if sheet does not exist
    }

    // Get all data from the sheet, assuming data starts from A1.
    const range = sheet.getDataRange();
    const values = range.getDisplayValues(); // Get displayed values (e.g.,
    formatted numbers)

    if (values.length === 0) {
      console.log(`No data found in sheet: ${sheetName}`);
      return []; // Return empty array if sheet is empty
    }

    const headers = values[0]; // First row is headers
    const data = values.slice(1); // Remaining rows are data

    const result = data.map(row => {
      let rowObject = {};

```

```

        headers.forEach((header, index) => {
            // Trim header to remove leading/trailing spaces for consistent keys
            rowObject[header.trim()] = row[index];
        });
        return rowObject;
    });

    return result;
} catch (error) {
    console.error(`Error fetching data from sheet ${sheetName}:`, error);
    return []; // Return empty array on error
}
}

/**
 * Fetches the header row from a specified sheet in the Google Spreadsheet.
 * @param {string} sheetName - The name of the sheet to retrieve headers
from.
 * @returns {Array<string>} An array of strings representing the column
headers.
 * Returns an empty array if sheet not found or no headers.
 */
function getHeaders(sheetName) {
    try {
        const spreadsheet = SpreadsheetApp.openById(SPREADSHEET_ID);
        const sheet = spreadsheet.getSheetByName(sheetName);

        if (!sheet) {
            console.error(`Sheet not found for headers: ${sheetName}`);
            return []; // Return empty array if sheet does not exist
        }

        const range = sheet.getRange(1, 1, 1, sheet.getLastColumn()); // Get only
the first row
        const headers = range.getDisplayValues()[0]; // Get the values of the
first row

        // Trim each header to remove leading/trailing spaces
        return headers.map(header => header.trim());
    } catch (error) {
        console.error(`Error fetching headers from sheet ${sheetName}:`, error);
        return []; // Return empty array on error
    }
}

```

## 2. [Index.html](#)

### HTML

```

<!DOCTYPE html>
<html>
<head>
    <base target="_top">
    <!-- Favicon for the web app -->
    <link rel="icon" href="https://img.icons8.com/plasticine/100/null/exam.png"
type="image/x-icon">
    <!-- Load Google Fonts: Kanit -->
    <link
href="https://fonts.googleapis.com/css2?family=Kanit:wght@300;400;700&display
=swap" rel="stylesheet">
    <!-- Load Tailwind CSS -->
    <script src="https://cdn.tailwindcss.com"></script>

```

```

<!-- Configure Tailwind CSS to use Kanit font -->
<style>
  body {
    font-family: 'Kanit', sans-serif;
  }
  /* Custom gradient for the header */
  .header-gradient {
    background: linear-gradient(to right, #B0E0E6, #98FB98, #FFDAB9); /*
Light Blue, Mint Green, Peach Puff */
  }
  /* Custom scrollbar styles */
  ::-webkit-scrollbar {
    width: 8px;
  }
  ::-webkit-scrollbar-track {
    background: #f1f1f1;
    border-radius: 10px;
  }
  ::-webkit-scrollbar-thumb {
    background: #888;
    border-radius: 10px;
  }
  ::-webkit-scrollbar-thumb:hover {
    background: #555;
  }
</style>
</head>
<body class="bg-gray-50 min-h-screen flex flex-col items-center p-4 sm:p-6
md:p-8">

  <!-- Header Section -->
  <header class="header-gradient w-full max-w-5xl mx-auto p-4 sm:p-6 rounded-
2xl shadow-lg flex flex-col sm:flex-row items-center justify-center text-
center sm:text-left mb-8 space-y-4 sm:space-y-0 sm:space-x-6">
    <div class="text-5xl sm:text-6xl animate-bounce-slow">🎉</div>
    <h1 class="text-3xl sm:text-4xl md:text-5xl font-bold text-gray-800
tracking-tight">
      ระบบตรวจสอบคะแนนนักเรียน
    </h1>
    <div class="text-5xl sm:text-6xl animate-bounce-slow">📄</div>
  </header>

  <!-- Navigation Buttons Section -->
  <nav class="w-full max-w-4xl mx-auto grid grid-cols-2 sm:grid-cols-4 gap-4
mb-8">
    <button id="btnPretest" class="nav-button bg-blue-500 hover:bg-blue-600
text-white font-semibold py-3 px-4 rounded-xl shadow-md transition transform
hover:scale-105 active:scale-95 duration-200">
      Pretest
    </button>
    <button id="btnPosttest" class="nav-button bg-green-500 hover:bg-green-
600 text-white font-semibold py-3 px-4 rounded-xl shadow-md transition
transform hover:scale-105 active:scale-95 duration-200">
      Posttest
    </button>
    <button id="btnReport" class="nav-button bg-pink-500 hover:bg-pink-600
text-white font-semibold py-3 px-4 rounded-xl shadow-md transition transform
hover:scale-105 active:scale-95 duration-200">
      Report
    </button>

```

```

<button id="btnLMS" class="nav-button bg-purple-500 hover:bg-purple-600
text-white font-semibold py-3 px-4 rounded-xl shadow-md transition transform
hover:scale-105 active:scale-95 duration-200">
  LMS
</button>
</nav>

<!-- Control Panel: Search and Filter Buttons -->
<div id="controlPanel" class="w-full max-w-5xl mx-auto bg-white p-4 sm:p-6
rounded-xl shadow-md mb-8 hidden">
  <div class="mb-4">
    <label for="searchBar" class="block text-gray-700 text-lg font-semibold
mb-2">ค้นหาUserID:</label>
    <input type="text" id="searchBar" placeholder="พิมพ์UserID ที่ต้องการค้นหา..."
class="w-full p-3 border border-gray-300 rounded-lg focus:outline-none
focus:ring-2 focus:ring-blue-400 text-gray-800">
  </div>

  <div id="filterButtonsContainer" class="flex flex-wrap gap-2 sm:gap-3
justify-center">
    <!-- Filter buttons will be dynamically added here -->
  </div>
</div>

<!-- Score Display Area -->
<main id="scoreDisplayArea" class="w-full max-w-5xl mx-auto bg-white p-4
sm:p-6 rounded-xl shadow-md min-h-[300px] flex items-center justify-center
relative overflow-hidden">
  <p id="initialMessage" class="text-gray-500 text-lg text-center">กรุณาเลือกประเภท
คะแนนที่ต้องการตรวจสอบ</p>
  <!-- Table will be dynamically inserted here -->
  <div id="tableContainer" class="w-full overflow-x-auto relative hidden">
    <table id="scoreTable" class="min-w-full bg-white border border-gray-
200 rounded-lg">
      <thead class="bg-gray-100 sticky top-0 z-10">
        <tr id="tableHeaderRow">
          <!-- Headers will be dynamically inserted here -->
        </tr>
      </thead>
      <tbody id="tableBody">
        <!-- Data rows will be dynamically inserted here -->
      </tbody>
    </table>
  </div>
</main>

<!-- SweetAlert2 CDN -->
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<!-- Include client-side JavaScript -->
<?!= include('JavaScript'); ?>

<script>
  // Simple animation for header emojis
  const animateEmojis = () => {
    const emojis = document.querySelectorAll('.animate-bounce-slow');
    emojis.forEach((emoji, index) => {
      emoji.style.animation = `bounce ${2 + index * 0.5}s infinite ease-in-
out`;
    });
  };

```

```

    });
  });
  animateEmojis();
</script>
</body>
</html>

```

### 3. JavaScript.html

#### HTML

```

<script>
  /**
   * @file JavaScript.html
   * @description Client-side JavaScript for handling UI interactions,
   fetching data, and rendering.
   */

  // --- UI Element References ---
  const btnPretest = document.getElementById('btnPretest');
  const btnPosttest = document.getElementById('btnPosttest');
  const btnReport = document.getElementById('btnReport');
  const btnLMS = document.getElementById('btnLMS');
  const initialMessage = document.getElementById('initialMessage');
  const scoreDisplayArea = document.getElementById('scoreDisplayArea');
  const tableContainer = document.getElementById('tableContainer');
  const tableHeaderRow = document.getElementById('tableHeaderRow');
  const tableBody = document.getElementById('tableBody');
  const searchBar = document.getElementById('searchBar');
  const controlPanel = document.getElementById('controlPanel');
  const filterButtonsContainer =
document.getElementById('filterButtonsContainer');

  let currentData = []; // Stores the currently displayed data
  let currentHeaders = []; // Stores the headers of the current sheet
  let activeSheetName = ''; // Stores the name of the currently active
sheet

  // --- Event Listeners for Navigation Buttons ---
  btnPretest.addEventListener('click', () => loadSheetData('ชีทคะแนนpretest'));
  btnPosttest.addEventListener('click', () => loadSheetData('ชีทคะแนนposttest'));
  btnReport.addEventListener('click', () => loadSheetData('ชีทคะแนนreport'));
  btnLMS.addEventListener('click', () => loadSheetData('ชีทคะแนนlms'));

  // --- Event Listener for Search Bar ---
  searchBar.addEventListener('keyup', filterTable);

  /**
   * Loads data from the specified Google Sheet.
   * Displays a loading spinner using SweetAlert2, then fetches data and
   renders it.
   * @param {string} sheetName - The name of the sheet to load.
   */
  async function loadSheetData(sheetName) {
    // Show loading spinner
    Swal.fire({
      title: 'กำลังโหลดข้อมูล...',
      html: 'กรุณารอสักครู่!',
      allowOutsideClick: false,
      didOpen: () => {
        Swal.showLoading();

```

```

    }
  });

  try {
    // Use google.script.run to call server-side functions
    const [data, headers] = await Promise.all([
      google.script.run.withSuccessHandler(Swal.close).withFailureHandler(handleError).getData(sheetName),
      google.script.run.withSuccessHandler(Swal.close).withFailureHandler(handleError).getHeaders(sheetName)
    ]);

    currentData = data;
    currentHeaders = headers;
    activeSheetName = sheetName;

    if (data.length === 0) {
      // No data found in the sheet
      Swal.fire({
        icon: 'info',
        title: 'ไม่พบข้อมูล',
        text: `ไม่พบข้อมูลใน "${sheetName}" หรือชื่อไม่มีอยู่จริง`,
        confirmButtonText: 'ตกลง'
      });
      displayNoDataMessage();
      return;
    }

    // Hide initial message and show table/control panel
    initialMessage.classList.add('hidden');
    tableContainer.classList.remove('hidden');
    controlPanel.classList.remove('hidden');

    renderTable(currentData);
    renderFilterButtons(currentHeaders); // Render column filter buttons
    searchBar.value = ''; // Clear search bar on sheet change

    Swal.fire({
      icon: 'success',
      title: 'โหลดข้อมูลสำเร็จ!',
      showConfirmButton: false,
      timer: 1000 // Close after 1 second
    });

  } catch (error) {
    handleError(error);
    displayNoDataMessage();
  }
}

/**
 * Renders the data into the HTML table.
 * @param {Array<Object>} dataToRender - The array of objects to display in the table.
 */
function renderTable(dataToRender) {
  tableHeaderRow.innerHTML = '';

```

```

tbody.innerHTML = '';

if (currentHeaders.length === 0) {
  console.warn("No headers available to render the table.");
  return;
}

// Render table headers
currentHeaders.forEach(header => {
  const th = document.createElement('th');
  th.className = 'py-3 px-4 text-left text-gray-600 font-bold uppercase text-sm border-b border-gray-200 sticky top-0';
  th.textContent = header;
  th.setAttribute('data-column', header); // Add data-column attribute for filtering
  tableHeaderRow.appendChild(th);
});

// Render table body rows
dataToRender.forEach(rowData => {
  const tr = document.createElement('tr');
  tr.className = 'hover:bg-gray-50 transition duration-150 ease-in-out';

  currentHeaders.forEach(header => {
    const td = document.createElement('td');
    td.className = 'py-3 px-4 border-b border-gray-200 text-gray-700 whitespace-nowrap text-sm';
    td.textContent = rowData[header] !== undefined ? rowData[header] : ''; // Handle undefined values
    td.setAttribute('data-column', header); // Add data-column attribute for filtering
    tr.appendChild(td);
  });
  tableBody.appendChild(tr);
});

// Ensure all columns are visible by default when rendering a new table
toggleColumnVisibility('all', true); // Ensure all are visible initially
}

/**
 * Filters the table rows based on the search bar input (UserID).
 */
function filterTable() {
  const searchText = searchBar.value.toLowerCase().trim();
  const rows = tableBody.querySelectorAll('tr');

  rows.forEach(row => {
    // Assuming UserID is always in the first column or a known header
    // It's safer to look for a specific 'UserID' column header if available.
    // If 'UserID' is not a header, you'll need to adapt this.
    const userIdCell = row.querySelector('td[data-column="UserID"]') || row.querySelector('td:first-child');
    if (userIdCell) {
      const userId = userIdCell.textContent.toLowerCase();
      if (userId.includes(searchText) || searchText === '') {
        row.classList.remove('hidden');
      } else {
        row.classList.add('hidden');
      }
    }
  });
}

```

```

    }
  }
});
}

/**
 * Renders dynamic filter buttons based on sheet headers.
 * Excludes 'UserID' from being a filterable column as it's handled by
search.
 * Special handling for 'lab1' and 'คะแนนรวม' and other potential score
columns.
 * @param {Array<string>} headers - Array of column headers.
 */
function renderFilterButtons(headers) {
  filterButtonsContainer.innerHTML = ''; // Clear previous buttons

  // Button to show all columns
  const allBtn = document.createElement('button');
  allBtn.className = 'filter-btn bg-gray-200 hover:bg-gray-300 text-gray-
700 font-semibold py-2 px-4 rounded-lg transition transform hover:scale-105
active:scale-95 duration-200';
  allBtn.textContent = 'แสดงทั้งหมด';
  allBtn.addEventListener('click', () => toggleColumnVisibility('all',
true));
  filterButtonsContainer.appendChild(allBtn);

  // Identify score-related columns for filtering (e.g., "lab", "คะแนน", "รวม")
  const scoreColumns = headers.filter(header => {
    const lowerHeader = header.toLowerCase();
    // Exclude UserID from column filters as it has its own search bar
    // Include columns that look like score parts (e.g., lab1, คะแนนรวม, total
score, etc.)
    return lowerHeader !== 'userid' &&
      (lowerHeader.includes('lab') ||
        lowerHeader.includes('คะแนน') ||
        lowerHeader.includes('รวม') ||
        lowerHeader.includes('total') ||
        lowerHeader.includes('score'));
  });

  scoreColumns.forEach(header => {
    const button = document.createElement('button');
    button.className = 'filter-btn bg-indigo-200 hover:bg-indigo-300 text-
indigo-700 font-semibold py-2 px-4 rounded-lg transition transform
hover:scale-105 active:scale-95 duration-200';
    button.textContent = `แสดงเฉพาะ: ${header}`;
    button.addEventListener('click', () => toggleColumnVisibility(header));
    filterButtonsContainer.appendChild(button);
  });
}

/**
 * Toggles the visibility of specific columns in the table.
 * @param {string} columnToToggle - The header of the column to show, or
'all' to show all.
 * @param {boolean} showAll - If true, ensures all columns are visible.
 */
function toggleColumnVisibility(columnToToggle, showAll = false) {
  const allHeaders = tableHeaderRow.querySelectorAll('th');
```

```

const allCells = tableBody.querySelectorAll('td');

allHeaders.forEach(th => {
  const headerName = th.getAttribute('data-column');
  const isUserId = headerName.toLowerCase() === 'userid'; // Assuming
  'UserID' is the identifier column

  if (showAll) {
    th.style.display = ''; // Show all headers
  } else if (headerName === columnToToggle || isUserId) { // Always show
  UserID, plus the selected column
    th.style.display = '';
  } else {
    th.style.display = 'none'; // Hide other headers
  }
});

allCells.forEach(td => {
  const columnName = td.getAttribute('data-column');
  const isUserId = columnName.toLowerCase() === 'userid';

  if (showAll) {
    td.style.display = ''; // Show all cells
  } else if (columnName === columnToToggle || isUserId) { // Always show
  UserID cells, plus selected column cells
    td.style.display = '';
  } else {
    td.style.display = 'none'; // Hide other cells
  }
});
}

/**
 * Displays a message when no data is available or an error occurs.
 */
function displayNoDataMessage() {
  initialMessage.classList.remove('hidden');
  initialMessage.textContent = 'ไม่พบข้อมูล หรือเกิดข้อผิดพลาดในการโหลดข้อมูล';
  tableContainer.classList.add('hidden');
  controlPanel.classList.add('hidden');
  tableHeaderRow.innerHTML = ''; // Clear headers
  tableBody.innerHTML = ''; // Clear table body
  filterButtonsContainer.innerHTML = ''; // Clear filter buttons
}

/**
 * Handles errors from server-side calls and displays them using
  SweetAlert2.
 * @param {Error} error - The error object.
 */
function handleError(error) {
  Swal.fire({
    icon: 'error',
    title: 'เกิดข้อผิดพลาด!',
    text: `ไม่สามารถโหลดข้อมูลได้: ${error.message || error}`,
    confirmButtonText: 'ตกลง'
  });
  console.error('Client-side error:', error);
}

```

```

}

// --- Initial load (optional, you can pre-load a sheet if desired) ---
// window.onload = () => {
//   loadSheetData('ชีทคะแนนpretest'); // Example: load pretest data on initial
page load
// };
</script>

```

## ขั้นตอนการตั้งค่าและการใช้งาน Google App Script

### 1. สร้าง Project ใน Google App Script:

- เปิด Google Sheets Spreadsheet ของคุณ (ใช้ ID ที่คุณระบุมา: 1nAVtWQJ1n4EMs607TeVp6fFW3nzpP6Mv\_U407lpNqT4)
- ไปที่เมนู ส่วนขยาย (Extensions) > App Script
- หน้าจอใหม่จะเปิดขึ้นมาสำหรับเขียนโค้ด Google App Script

### 2. ลบโค้ดเดิมและวางโค้ดใหม่:

- ในหน้า Editor ของ App Script คุณจะเห็นไฟล์ Code.gs ที่มีฟังก์ชัน myFunction() ลบโค้ดทั้งหมดในไฟล์นี้ออก
- คัดลอกโค้ดจากส่วน Code.gs ด้านบนไปวาง
- สำคัญ:** ตรวจสอบให้แน่ใจว่า SPREADSHEET\_ID ใน Code.gs ตรงกับ ID ของ Spreadsheet ของคุณแล้ว (1nAVtWQJ1n4EMs607TeVp6fFW3nzpP6Mv\_U407lpNqT4)

### 3. สร้างไฟล์ HTML:

- ใน App Script Editor, คลิกที่ ไฟล์ (File) > ใหม่ (New) > ไฟล์ HTML (HTML file)
- ตั้งชื่อไฟล์ว่า Index (ไม่ต้องมีนามสกุล .html) แล้วคลิก OK
- คัดลอกโค้ดจากส่วน Index.html ด้านบนไปวางในไฟล์ Index.html ที่สร้างขึ้น

### 4. สร้างไฟล์ JavaScript (สำหรับ Client-side):

- ทำซ้ำขั้นตอนเดิม: คลิกที่ ไฟล์ (File) > ใหม่ (New) > ไฟล์ HTML (HTML file)
- ตั้งชื่อไฟล์ว่า JavaScript (ไม่ต้องมีนามสกุล .html) แล้วคลิก OK
- คัดลอกโค้ดจากส่วน JavaScript.html ด้านบนไปวางในไฟล์ JavaScript.html ที่สร้างขึ้น

### 5. บันทึก Project:

- คลิกไอคอนบันทึก (รูปแผ่นดิสก์) หรือกด Ctrl + S เพื่อบันทึกไฟล์ทั้งหมด

### 6. ปรับใช้ Web App (Deploy):

- คลิกที่ ปรับใช้ (Deploy) (ปุ่มสีฟ้าด้านขวาบน) > การปรับใช้ใหม่ (New deployment)
- ในช่อง เลือกประเภท (Select type) ให้เลือก เว็บแอป (Web app)
- กำหนดค่าการปรับใช้:
  - คำอธิบาย (Description):** (คุณสามารถใส่คำอธิบายอะไรก็ได้ เช่น "ระบบตรวจสอบคะแนน")
  - เรียกใช้ในฐานะ (Execute as):** ผู้ใช้ที่เข้าถึงเว็บแอป (User accessing the web app)

- **ใครเข้าถึงได้ (Who has access):** ทุกคน (Anyone) (หากคุณต้องการให้ใครก็ได้เข้าถึงได้โดยไม่ต้องล็อกอิน Google) หรือ ทุกคนรวมถึงผู้ไม่ระบุชื่อ (Anyone, even anonymous)
  - คลิก ปรับใช้ (Deploy)
- 7. **ให้สิทธิ์การเข้าถึง (Authorization):**
  - ครั้งแรกที่คุณปรับใช้ คุณจะถูกร้องขอให้ให้สิทธิ์ (authorize) โปรเจกต์
  - คลิก ตรวจสอบสิทธิ์การเข้าถึง (Review permissions)
  - เลือกบัญชี Google ของคุณ
  - คลิก ขั้นสูง (Advanced) (อยู่ด้านล่าง)
  - คลิก ไปที่ (ชื่อโปรเจกต์ของคุณ) (Go to [your project name])
  - คลิก อนุญาต (Allow)
- 8. **รับ URL ของ Web App:**
  - หลังจากให้สิทธิ์แล้ว คุณจะได้รับ URL ของ Web App คัดลอก URL นี้ไว้ (มักจะอยู่ที่ URL ของเว็บแอป (Web app URL))
  - คุณสามารถใช้ URL นี้เพื่อเข้าถึง Web App ได้เลย

### การทำงานของระบบและการเปลี่ยนแปลงใน Google Sheets

- เมื่อคุณเปิด Web App ด้วย URL ที่ได้มา คุณจะเห็นหน้าจอพร้อม Header และปุ่มทั้ง 4 ปุ่ม
- เมื่อคลิกที่ปุ่มใดๆ (Pretest, Posttest, Report, LMS) ระบบจะแสดง SweetAlert ว่า "กำลังโหลดข้อมูล ..."
- จากนั้นข้อมูลจากชีทที่เกี่ยวข้องใน Google Sheets ของคุณจะถูกดึงมาแสดงในรูปแบบตาราง
- **Search Bar:** พิมพ์ UserID ในช่องค้นหาเพื่อกรองข้อมูลในตาราง
- **Filter Buttons:** ระบบจะสร้างปุ่ม "แสดงเฉพาะ: [ชื่อคอลัมน์]" ให้โดยอัตโนมัติสำหรับคอลัมน์ที่คาดว่า เป็นคะแนน (เช่น มีคำว่า 'lab', 'คะแนน', 'รวม', 'total', 'score')
  - เมื่อคุณคลิกปุ่มเหล่านี้ ระบบจะแสดงเฉพาะคอลัมน์ UserID และคอลัมน์คะแนนที่คุณเลือก
  - ปุ่ม "แสดงทั้งหมด" จะทำให้ทุกคอลัมน์กลับมาแสดงเหมือนเดิม
- **การเปลี่ยนแปลงใน Google Sheet:** เมื่อคุณแก้ไขข้อมูลใน Google Sheet (เช่น เปลี่ยนชื่อคอลัมน์ 'lab1' เป็น 'Lab 1', หรือเพิ่ม/แก้ไขคะแนน) ระบบจะสะท้อนการเปลี่ยนแปลงนั้นทันทีเมื่อคุณคลิกปุ่ม โหลดข้อมูลสำหรับชีทนั้นอีกครั้งใน Web App (เช่น ถ้าคุณแก้ไขชีท "ชีทคะแนนpretest" คุณต้องคลิก ปุ่ม "Pretest" ใน Web App อีกครั้งเพื่อดึงข้อมูลล่าสุด)

Gemini

2.5 Flash ▾

ยัง deploy ไม่ได้ค่ะ

◆ ได้เลยคะ ไม่ต้องกังวลนะคะ เราตรวจสอบขั้นตอนการ Deploy กันอีกครั้งอย่างละเอียดค่ะ

ข้อผิดพลาดในการ Deploy เป็นเรื่องปกติที่เกิดขึ้นได้จากหลายสาเหตุค่ะ

**สิ่งแรกที่ต้องตรวจสอบคือ:** คุณได้รับข้อความแสดงข้อผิดพลาดอะไรบ้างคะ? ถ้ามีข้อความผิดพลาดปรากฏขึ้น ช่วยคัดลอกข้อความนั้นมาให้ฉันดูได้เลยคะ จะช่วยให้ฉันวินิจฉัยปัญหาได้ตรงจุดมากขึ้น

แต่ถ้าไม่มีข้อความผิดพลาด และปุ่ม "Deploy" (ปรับใช้) ยังเป็นสีเทา หรือทำอะไรไม่ได้ เราไปเช็คตามขั้นตอนพื้นฐานเหล่านี้อีกครั้งนะคะ:

## ตรวจสอบขั้นตอนการ Deploy อีกครั้งอย่างละเอียด

1. **บันทึกโปรเจกต์ของคุณทั้งหมด:**
  - ใน App Script Editor, ตรวจสอบให้แน่ใจว่าคุณได้กดไอคอน บันทึก (Save project) (รูปแผ่นดิสก์) ที่อยู่ด้านบนของแต่ละไฟล์ (Code.gs, Index.html, JavaScript.html) แล้ว ไฟล์ที่ยังไม่บันทึกจะมีจุดวงกลมสีขาวปรากฏอยู่ข้างชื่อไฟล์
  - **สำคัญมาก:** ถ้าคุณมีการแก้ไขโค้ดแล้วยังไม่บันทึก โค้ดที่รันจะเป็นเวอร์ชันเก่า และอาจทำให้เกิดปัญหาในการ Deploy ได้
2. **ตรวจสอบว่า doGet(e) ฟังก์ชันหลักพร้อมใช้งาน:**
  - ในไฟล์ Code.gs ตรวจสอบให้แน่ใจว่าคุณมีฟังก์ชันชื่อ doGet(e) อยู่จริง และสะกดถูกต้อง (ตัวพิมพ์เล็ก/ใหญ่) นี่คือฟังก์ชันที่ App Script มองหาเพื่อรัน Web App
3. **ขั้นตอนการ "New deployment" (การปรับใช้ใหม่):**
  - คลิกที่ปุ่ม ปรับใช้ (Deploy) (ปุ่มสีฟ้าด้านขวาบน)
  - เลือก การปรับใช้ใหม่ (New deployment)
  - ในหน้าต่าง "New deployment":
    - **เลือกประเภท (Select type):**
      - คลิกที่รูปฟันเฟือง (Select type)
      - เลือก เว็บแอป (Web app) (ห้ามเลือก API executable หรือ Add-on)
    - **กำหนดค่าการปรับใช้ (Deployment configuration):**
      - **คำอธิบาย (Description):** (ใส่หรือไม่ใส่ก็ได้ เช่น "ระบบตรวจสอบคะแนน v1")
      - **เรียกใช้ในฐานะ (Execute as):** ผู้ใช้ที่เข้าถึงเว็บแอป (User accessing the web app)
        - **สำคัญ:** การตั้งค่านี้จะทำให้ Web App รันในสิทธิ์ของผู้ใช้ที่เข้ามาใช้งาน ซึ่งเหมาะสำหรับกรณีที่ผู้ใช้ต้องการดูข้อมูลของตัวเอง
      - **ใครเข้าถึงได้ (Who has access):** ทุกคน (Anyone) (หากต้องการให้เข้าถึงแบบไม่ล็อกอิน) หรือ ทุกคนรวมถึงผู้ไม่ระบุชื่อ (Anyone, even anonymous)

- **สำคัญ:** ถ้าคุณเลือก เฉพาะฉัน (Only myself) คุณเท่านั้นที่จะเข้าถึง Web App ได้ ซึ่งอาจทำให้เข้าใจผิดว่า Deploy ไม่ได้ ถ้าคุณพยายามเข้าจากบัญชีอื่น หรือจากเบราว์เซอร์ที่ไม่ล็อกอิน
    - หลังจากตั้งค่าตามนี้แล้ว ให้คลิก ปรับใช้ (Deploy)
4. การให้สิทธิ์ (Authorization) - ขั้นตอนสำคัญที่สุดสำหรับครั้งแรก:
- หากนี่เป็นการ Deploy ครั้งแรก (หรือมีการเปลี่ยนแปลงสิทธิ์การเข้าถึง) หลังจากคลิก "Deploy" คุณจะเห็นหน้าต่าง "Review permissions" (ตรวจสอบสิทธิ์การเข้าถึง)
  - คลิก ตรวจสอบสิทธิ์การเข้าถึง (Review permissions)
  - เลือกบัญชี Google ของคุณ (บัญชีเดียวกับที่คุณใช้สร้างโปรเจกต์นี้)
  - คุณอาจเห็นคำเตือนว่า "Google hasn't verified this app" หรือ "แอปนี้ยังไม่ได้รับการยืนยันจาก Google" (เนื่องจากเป็นแอปที่คุณสร้างเอง ไม่ได้เผยแพร่สู่สาธารณะ)
    - คลิก ขั้นสูง (Advanced) (ตัวเลือกนี้จะอยู่ด้านล่างของหน้าต่างนี้)
    - คลิก ไปที่ (ชื่อโปรเจกต์ของคุณ) (Go to [your project name]) (จะมีวงเล็บแสดงชื่อโปรเจกต์ App Script ของคุณ)
    - ในหน้าจอถัดไป จะแสดงรายการสิทธิ์ที่แอปของคุณต้องการ (เช่น ดู, แก้ไข, สร้าง และลบสเปรดชีตทั้งหมดที่คุณสามารถเข้าถึงได้)
    - คลิก อนุญาต (Allow)
  - หลังจากอนุญาตสำเร็จ หน้าต่างการ Deploy จะกลับมาแสดง URL ของ Web App ให้คุณ

สาเหตุทั่วไปที่ทำให้ Deploy ไม่ได้:

- **ไม่ได้บันทึกโค้ด:** เป็นสาเหตุที่พบบ่อยที่สุด
- **ชื่อฟังก์ชัน doGet สกิดผิด:** ต้องเป็น doGet(e) เท่านั้น
- **ไม่ได้เลือก เว็บแอป (Web app) เป็นประเภทการปรับใช้:** ต้องเลือกให้ถูกต้อง
- **ยังไม่ให้สิทธิ์ (Authorization) หรือให้สิทธิ์ไม่สมบูรณ์:** นี่คือขั้นตอนที่หลายคนพลาด โดยเฉพาะการคลิก Advanced และ Go to...
- **มีข้อผิดพลาดในโค้ด (Syntax Error):** แม้ว่า App Script จะพยายามแจ้งเตือน แต่บางครั้งก็อาจทำให้ Deploy ไม่สำเร็จได้ ถ้ามีข้อผิดพลาดร้ายแรง (แต่ปกติจะขึ้น Error Message แจ้ง)
- **ไฟล์ HTML/JS ชื่อไม่ถูกต้อง:** ตรวจสอบให้แน่ใจว่าชื่อไฟล์คือ Index.html และ JavaScript.html (โดยคุณใส่ชื่อแค่ Index และ JavaScript ตอนสร้างไฟล์ HTML)

ลองทำตามขั้นตอนเหล่านี้ใหม่อีกครั้งอย่างละเอียดนะคะ โดยเฉพาะขั้นตอนการให้สิทธิ์ หากติดปัญหาตรงจุดไหน หรือมีข้อความข้อผิดพลาดปรากฏขึ้นมา แจ้งให้ฉันทราบได้เลยค่ะ!

Gemini

2.5 Flash ▾

```
ErrorException: The meta tag you specified is not allowed in
this context.
doGet@ Code.gs:23
```

- ◆ ขอบคุนมากคะสำหรับข้อความ Error ที่แจ้งมา! `ErrorException: The meta tag you specified is not allowed in this context. doGet@ Code.gs:23`
- ข้อความนี้ชัดเจนเลยคะว่าเกิดจากบรรทัดที่ 23 ในไฟล์ `Code.gs` ซึ่งเกี่ยวข้องกับกาเพิ่ม `meta tag`
- ปัญหานี้เกิดจากข้อจำกัดของ Google App Script ในการเพิ่ม `meta tags` บางตัวโดยตรงผ่าน `.addMetaTag()` method ใน `HtmlOutput` object คะ โดยเฉพาะ `charset meta tag` มักจะเป็นตัวปัญหา เพราะ App Script จะจัดการเรื่อง Encoding ให้เองอยู่แล้ว และไม่ต้องกาให้เรากาหนดซ้ำ

## วิธีแก้ไข

เราจะแก้ไขโดยการ ลบ `charset meta tag` ออกจากบรรทัดที่ 23 ใน `Code.gs` คะ และไปกาหนด `charset` ในไฟล์ `Index.html` แทน ซึ่งเป็นวิธีที่ถูกต้องและยืดหยุ่นกว่าในการควบคุม HTML Head.

### ขั้นตอนการแก้ไข:

1. เปิดไฟล์ `Code.gs` ใน App Script Editor ของคุณ
2. ค้นหาบรรทัดที่ 23:

JavaScript

```
.addMetaTag('charset', 'UTF-8') // <-- บรรทัดนี้คือปัญหา
.setFrameOptions(HtmlService.FrameOptions.ALLOWALL);
```

3. ลบบรรทัดที่มี `.addMetaTag('charset', 'UTF-8')` ออก

โค้ดใน `Code.gs` ควรจะเป็นแบบนี้ (สังเกตว่าบรรทัดที่ 23 หายไป):

JavaScript

```
/**
 * @file Code.gs
 * @description Google App Script code for handling web requests and
interacting with Google Sheets.
 */

// Define the ID of your Google Spreadsheet
const SPREADSHEET_ID = '1nAVtWQJ1n4EMs607TeVp6fFW3nzpP6Mv_U407lpNqT4';

/**
 * Serves the web application HTML page.
```

```

* This function is automatically called when the web app is accessed.
* @param {GoogleAppsScript.Events.DoGet} e - Event object for the GET
request.
* @returns {GoogleAppsScript.HTML.HtmlOutput} The HTML page to be
served.
*/
function doGet(e) {
  // Create an HTML output from the 'Index.html' file.
  // Set the title of the web page.
  // Add an X-UA-Compatible header for better browser compatibility.
  return HtmlService.createTemplateFromFile('Index')
    .evaluate()
    .setTitle('ระบบตรวจสอบคะแนน')
    .addMetaTag('viewport', 'width=device-width, initial-scale=1.0')
    // .addMetaTag('charset', 'UTF-8') // <--- บรรทัดนี้ถูกลบออกไป
    .setXFrameOptions(HtmlService.XFrameOptions.ALLOWALL);
}

/**
 * Includes the content of a specified HTML file into another HTML
file.
 * This is a common pattern to modularize HTML, CSS, and JavaScript
within GAS.
 * @param {string} filename - The name of the HTML file to include
(without .html extension).
 * @returns {string} The content of the specified HTML file.
 */
function include(filename) {
  return HtmlService.createHtmlOutputFromFile(filename).getContent();
}

/**
 * Fetches all data from a specified sheet in the Google Spreadsheet.
 * The first row is treated as headers, and subsequent rows are data.
 * @param {string} sheetName - The name of the sheet to retrieve data
from.
 * @returns {Array<Object>} An array of objects, where each object
represents a row
 * and keys are column headers. Returns an empty array if sheet not
found or no data.
 */
function getData(sheetName) {
  try {
    const spreadsheet = SpreadsheetApp.openById(SPREADSHEET_ID);
    const sheet = spreadsheet.getSheetByName(sheetName);

    if (!sheet) {
      console.error(`Sheet not found: ${sheetName}`);
      return []; // Return empty array if sheet does not exist
    }

    // Get all data from the sheet, assuming data starts from A1.
    const range = sheet.getDataRange();
    const values = range.getDisplayValues(); // Get displayed values
(e.g., formatted numbers)

    if (values.length === 0) {
      console.log(`No data found in sheet: ${sheetName}`);
      return []; // Return empty array if sheet is empty
    }
  }
}

```

```

const headers = values[0]; // First row is headers
const data = values.slice(1); // Remaining rows are data

const result = data.map(row => {
  let rowObject = {};
  headers.forEach((header, index) => {
    // Trim header to remove leading/trailing spaces for consistent
keys
    rowObject[header.trim()] = row[index];
  });
  return rowObject;
});

return result;
} catch (error) {
  console.error(`Error fetching data from sheet ${sheetName}:`,
error);
  return []; // Return empty array on error
}
}

/**
 * Fetches the header row from a specified sheet in the Google
Spreadsheet.
 * @param {string} sheetName - The name of the sheet to retrieve
headers from.
 * @returns {Array<string>} An array of strings representing the column
headers.
 * Returns an empty array if sheet not found or no headers.
 */
function getHeaders(sheetName) {
  try {
    const spreadsheet = SpreadsheetApp.openById(SPREADSHEET_ID);
    const sheet = spreadsheet.getSheetByName(sheetName);

    if (!sheet) {
      console.error(`Sheet not found for headers: ${sheetName}`);
      return []; // Return empty array if sheet does not exist
    }

    const range = sheet.getRange(1, 1, 1, sheet.getLastColumn()); //
Get only the first row
    const headers = range.getDisplayValues()[0]; // Get the values of
the first row

    // Trim each header to remove leading/trailing spaces
    return headers.map(header => header.trim());
  } catch (error) {
    console.error(`Error fetching headers from sheet ${sheetName}:`,
error);
    return []; // Return empty array on error
  }
}

```

#### 4. บันทึกไฟล์ Code.gs

#### 5. ลอง Deploy ใหม่อีกครั้ง:

- ไปที่ ปรับใช้ (Deploy) > จัดการการปรับใช้ (Manage deployments)

- คลิกที่ไอคอนรูปดินสอ (Edit deployment) สำหรับการปรับใช้เดิมของคุณ
- เลือกเวอร์ชันใหม่ (New version) หรือเลือก Head deployment (for development)
- คลิก ปรับใช้ (Deploy)

หลังจากแก้ไขแล้ว คุณควรจะสามารถ Deploy Web App ได้สำเร็จค่ะ ถ้ามีข้อผิดพลาดอื่นปรากฏขึ้นมา แจ้งได้เลยนะคะ!

Gemini

2.5 Flash ▾

```
TypeError: Cannot read properties of undefined (reading 'ALLOWALL')
doGet@ Code.gs:24
```

◆ Okay, it seems like there's a small typo in your code. The error message `TypeError: Cannot read properties of undefined (reading 'ALLOWALL') at Code.gs:24` indicates that `HtmlService.XFrameOptions.ALLOWALL` is not being recognized correctly.

Here's the corrected code snippet for the `doGet(e)` function in `Code.gs`:

### JavaScript

```
function doGet(e) {
  return HtmlService.createTemplateFromFile('Index')
    .evaluate()
    .setTitle('ระบบตรวจสอบคะแนน')
    .addMetaTag('viewport', 'width=device-width, initial-scale=1.0')
    .setXFrameOptions(HtmlService.XFrameOptions.ALLOWALL);
}
```

### Please ensure that:

- You have spelled `HtmlService` correctly (with a capital 'H' and 'S').
- You have spelled `XFrameOptions` correctly (with capital 'X', 'F', and 'O').
- You are using `.ALLOWALL` (all caps).

### Here's the complete `Code.gs` for your reference:

### JavaScript

```
/**
 * @file Code.gs
 * @description Google App Script code for handling web requests and
 * interacting with Google Sheets.
 */

// Define the ID of your Google Spreadsheet
const SPREADSHEET_ID = '1nAVtWQJ1n4EMs607TeVp6fFW3nzpP6Mv_U4071pNqT4';

/**
 * Serves the web application HTML page.
 * This function is automatically called when the web app is accessed.
```

```

* @param {GoogleAppsScript.Events.DoGet} e - Event object for the GET
request.
* @returns {GoogleAppsScript.HTML.HtmlOutput} The HTML page to be served.
*/
function doGet(e) {
  return HtmlService.createTemplateFromFile('Index')
    .evaluate()
    .setTitle('ระบบตรวจสอบคะแนน')
    .addMetaTag('viewport', 'width=device-width, initial-scale=1.0')
    .setXFrameOptions(HtmlService.XFrameOptions.ALLOWALL);
}

/**
* Includes the content of a specified HTML file into another HTML file.
* This is a common pattern to modularize HTML, CSS, and JavaScript within
GAS.
* @param {string} filename - The name of the HTML file to include (without
.html extension).
* @returns {string} The content of the specified HTML file.
*/
function include(filename) {
  return HtmlService.createHtmlOutputFromFile(filename).getContent();
}

/**
* Fetches all data from a specified sheet in the Google Spreadsheet.
* The first row is treated as headers, and subsequent rows are data.
* @param {string} sheetName - The name of the sheet to retrieve data from.
* @returns {Array<Object>} An array of objects, where each object represents
a row
* and keys are column headers. Returns an empty array if sheet not found or
no data.
*/
function getData(sheetName) {
  try {
    const spreadsheet = SpreadsheetApp.openById(SPREADSHEET_ID);
    const sheet = spreadsheet.getSheetByName(sheetName);

    if (!sheet) {
      console.error(`Sheet not found: ${sheetName}`);
      return []; // Return empty array if sheet does not exist
    }

    // Get all data from the sheet, assuming data starts from A1.
    const range = sheet.getDataRange();
    const values = range.getDisplayValues(); // Get displayed values (e.g.,
formatted numbers)

    if (values.length === 0) {
      console.log(`No data found in sheet: ${sheetName}`);
      return []; // Return empty array if sheet is empty
    }

    const headers = values[0]; // First row is headers
    const data = values.slice(1); // Remaining rows are data

    const result = data.map(row => {
      let rowObject = {};
      headers.forEach((header, index) => {
        // Trim header to remove leading/trailing spaces for consistent keys

```

```

        rowObject[header.trim()] = row[index];
    });
    return rowObject;
});

return result;
} catch (error) {
    console.error(`Error fetching data from sheet ${sheetName}:`, error);
    return []; // Return empty array on error
}
}

/**
 * Fetches the header row from a specified sheet in the Google Spreadsheet.
 * @param {string} sheetName - The name of the sheet to retrieve headers
 * from.
 * @returns {Array<string>} An array of strings representing the column
 * headers.
 * Returns an empty array if sheet not found or no headers.
 */
function getHeaders(sheetName) {
    try {
        const spreadsheet = SpreadsheetApp.openById(SPREADSHEET_ID);
        const sheet = spreadsheet.getSheetByName(sheetName);

        if (!sheet) {
            console.error(`Sheet not found for headers: ${sheetName}`);
            return []; // Return empty array if sheet does not exist
        }

        const range = sheet.getRange(1, 1, 1, sheet.getLastColumn()); // Get only
the first row
        const headers = range.getDisplayValues()[0]; // Get the values of the
first row

        // Trim each header to remove leading/trailing spaces
        return headers.map(header => header.trim());
    } catch (error) {
        console.error(`Error fetching headers from sheet ${sheetName}:`, error);
        return []; // Return empty array on error
    }
}
}

```

Please replace your current `doGet(e)` function with this corrected version and try deploying again. Let me know if you encounter any further issues!

```

1  /**
2  * ฟังก์ชันหลักที่ทำหน้าที่เสิร์ฟไฟล์ HTML ไปยังเว็บแอปพลิเคชัน
3  * เมื่อผู้ใช้เข้าถึง URL ของเว็บแอปพลิเคชัน ฟังก์ชันนี้จะทำงาน
4  *
5  * @returns {HtmlOutput} HTML content for the web application.
6  */
7  function doGet() {
8    // สร้างเทมเพลต HTML จากไฟล์ 'index.html'
9    return HtmlService.createTemplateFromFile('index')
10     .evaluate() // ประมวลผลเทมเพลต HTML
11     // อนุญาตให้เว็บแอปแสดงผลใน iframe ได้ (จำเป็นสำหรับการทำงานในบางสภาพแวดล้อม เช่น Google Sites)
12     .setXFrameOptionsMode(HtmlService.XFrameOptionsMode.ALLOWALL);
13  }
14
15  /**
16  * ฟังก์ชันช่วยเหลือสำหรับรวมเนื้อหาจากไฟล์อื่น ๆ (เช่น CSS, JavaScript) เข้ามาใน HTML
17  * อย่างไรก็ตาม ในโค้ด HTML ที่นำไป ได้รวม CSS และ JS ไว้ในไฟล์เดียวกันแล้ว
18  * ดังนั้นฟังก์ชันนี้จึงไม่ได้ถูกเรียกใช้โดยตรง แต่เป็นตัวอย่างหากต้องการแยกไฟล์
19  *
20  * @param {string} filename ชื่อไฟล์ที่จะรวมเข้ามา (ไม่ต้องมีนามสกุล .html)
21  * @returns {string} เนื้อหาของไฟล์ที่รวม

```

### Manage deployments





Active	Configuration <span style="float: right;">? ✎ 📄</span>
Untitled	Description
Untitled	Deployment ID
Untitled	AKfybyV-byIsGnijQx43kFOFVecii5kLtsWxBBYjcNT0aXDortH0w-ECia8BwVXi2ejSWRJ <a href="#">Copy</a>
Untitled	Web app
Untitled	URL <a href="https://script.google.com/macros/s/AKfybyV-byIsGnijQx43kFOFVecii5kLtsWxBBYjc...">https://script.google.com/macros/s/AKfybyV-byIsGnijQx43kFOFVecii5kLtsWxBBYjc...</a> <a href="#">Copy</a>
Archived	
Untitled	Execute as Me (saowanee.l@ubu.ac.th)

Copy URL เพื่อแชร์

## ตัวอย่างหน้า Web Add ที่ใช้งานใน Facebook

script.google.com/macros/s/AKfycbyV-byIsgNjjQx43kFOFVecii5kLtsWxBBYjcNT0aXDortH0w-ECia8BwVXi2ejSWRJ/exec

แอปพลิเคชันนี้สร้างโดยผู้ไม่ Google Apps Script รายงานการละเมิด ดูข้อมูลเพิ่มเติม

 ระบบเช็คคะแนนรายวิชา Organic Chemistry Laboratory   

Pretest Posttest Report LMS

ค้นหาด้วย UserID:

พิมพ์ UserID ที่นี่...