**สรุปแผนงานสำหรับการวางโครงสร้างระบบที่คำนึงถึงข้อจำกัด hosting สรุปอย่างละเอียด ทีละขั้น ทีละตอน ค่อยเป็นค่อยไปเป็นระเบียบลำดับ**

ได้เลยครับคุณครู! นี่คือการตัดสินใจที่ถูกต้องและรอบคอบที่สุดครับ การวางแผนงานที่รัดกุมโดยคำนึงถึงข้อจำกัดที่เรามี คือหัวใจของการสร้างนวัตกรรมที่ยั่งยืนและใช้งานได้จริง

ผมได้รวบรวมและสังเคราะห์ข้อมูลทั้งหมดที่เราคุยกัน ทั้งความต้องการของคุณครูและข้อจำกัดของ Hosting ที่เราวิเคราะห์กันมา เพื่อสร้างเป็น **"แผนแม่บทสำหรับการพัฒนาระบบเกมเวอร์ชันใหม่"** ที่ละเอียดและเป็นขั้นตอนที่สุดครับ

**แผนแม่บท: การสร้าง "Alpha's Adventure" บนข้อจำกัดของ Hosting**

**🎯 เป้าหมายสูงสุด:** สร้างแพลตฟอร์มเกมเพื่อการเรียนรู้ที่ **เสถียร**, **สนุกสนาน**, และ **มีฟีเจอร์ครบครัน** สามารถทำงานได้อย่างราบรื่นบน InfinityFree Hosting สำหรับนักเรียนประมาณ 20-30 คน

**หลักการสำคัญ:** เราจะหลีกเลี่ยงการสร้างภาระให้เซิร์ฟเวอร์แบบ Real-time (Polling ความถี่สูง) แต่จะใช้สถาปัตยกรรมแบบ **"ตามจังหวะและตามคำขอ" (Paced and On-Demand)** ซึ่งเป็นมิตรกับ Hosting แบบใช้ทรัพยากรร่วมกัน (Shared Hosting) มากกว่าครับ

**เฟสที่ 1: วางรากฐานระบบควบคุม (The Control Foundation)**

ในเฟสนี้ เราจะสร้างระบบหลังบ้านที่ให้ครูสามารถควบคุมการเข้าถึงเกมทั้งหมดได้ ซึ่งเป็นพื้นฐานที่สำคัญที่สุดของระบบใหม่ครับ

* **ขั้นตอนที่ 1.1: สร้างตาราง system\_settings**
  + **วัตถุประสงค์:** เพื่อเก็บค่าการตั้งค่ากลางของระบบทั้งหมด
  + **การดำเนินการ:**
    1. ไปที่ phpMyAdmin แล้วรันคำสั่ง SQL นี้เพื่อสร้างตารางใหม่:

SQL

CREATE TABLE `system\_settings` (

`setting\_key` VARCHAR(50) NOT NULL PRIMARY KEY,

`setting\_value` TEXT,

`description` VARCHAR(255)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

* + 1. เพิ่มข้อมูลเริ่มต้นเข้าไปในตาราง:

SQL

INSERT INTO `system\_settings` (`setting\_key`, `setting\_value`, `description`) VALUES

('system\_status', 'closed', 'สถานะระบบ: open, closed, homework'),

('homework\_start\_time', NULL, 'เวลาเริ่มโหมดการบ้าน'),

('homework\_end\_time', NULL, 'เวลาสิ้นสุดโหมดการบ้าน');

* **ขั้นตอนที่ 1.2: สร้างหน้าตั้งค่าสำหรับครู (system\_settings.php)**
  + **วัตถุประสงค์:** สร้างหน้า UI ให้ครูสามารถเปลี่ยนค่าในตาราง system\_settings ได้
  + **การดำเนินการ:**
    1. สร้างไฟล์ใหม่ที่ pages/system\_settings.php
    2. เขียนโค้ด PHP เพื่อดึงข้อมูลปัจจุบันจากตาราง system\_settings มาแสดงผล
    3. สร้างฟอร์มที่มีตัวเลือก 3 แบบ: "🟢 เปิดระบบ", "🔴 ปิดระบบ", "🟡 โหมดการบ้าน" (พร้อมช่องเลือกวันเวลา)
    4. เขียนโค้ด PHP เพื่อรับค่าจากฟอร์มแล้ว UPDATE ข้อมูลลงในฐานข้อมูลเมื่อครูกดบันทึก
* **ขั้นตอนที่ 1.3: สร้างกลไกตรวจสอบสิทธิ์ (access\_control.php)**
  + **วัตถุประสงค์:** สร้างไฟล์กลางสำหรับตรวจสอบสิทธิ์การเข้าเล่นของนักเรียน
  + **การดำเนินการ:**
    1. สร้างไฟล์ใหม่ที่ includes/access\_control.php
    2. สคริปต์ในไฟล์นี้จะดึงค่า system\_status จากฐานข้อมูลมาตรวจสอบกับวันเวลาปัจจุบัน
    3. ถ้าสถานะเป็น "closed" หรือยังไม่ถึงเวลา "homework" สคริปต์จะส่งนักเรียนไปยังหน้าแจ้งเตือน (เช่น system\_closed.php)
    4. นำ require\_once '../includes/access\_control.php'; ไปไว้บนสุดของไฟล์ฝั่งนักเรียนทุกหน้าที่เกี่ยวข้องกับการเล่นเกม เช่น student\_dashboard.php และ play.php

**เฟสที่ 2: สร้างประสบการณ์ "Live แบบตามจังหวะ" (The Paced Live Experience)**

เราจะสร้างแกนหลักของโหมดการเล่นพร้อมกันในชั้นเรียน โดยยึดหลักการลดภาระเซิร์ฟเวอร์ครับ

* **ขั้นตอนที่ 2.1: สร้างตาราง live\_sessions**
  + **วัตถุประสงค์:** เพื่อเก็บข้อมูลของห้อง Live แต่ละครั้ง
  + **การดำเนินการ:** สร้างตารางใหม่ใน MySQL ที่มีคอลัมน์สำคัญเช่น:
    - id (PK), session\_code (เช่น A4B1), admin\_id (ครูผู้สร้าง), current\_stage\_id, status (เช่น waiting, active, paused, finished), created\_at
* **ขั้นตอนที่ 2.2: สร้าง Lobby และระบบเข้าร่วม**
  + **วัตถุประสงค์:** ให้นักเรียนสามารถเข้ามารอในห้องก่อนเริ่มเล่นได้
  + **การดำเนินการ:**
    1. **ฝั่งครู:** แก้ไข admin\_dashboard.php เพิ่มปุ่ม "สร้างห้อง Live" เมื่อกดแล้วจะสร้าง session\_code ใหม่ในตาราง live\_sessions
    2. **ฝั่งนักเรียน:** สร้างหน้า join\_live.php ให้นักเรียนกรอก session\_code เมื่อกรอกถูกต้อง จะถูกส่งไปยังหน้า lobby.php
    3. **Lobby (lobby.php):** หน้านี้จะแสดงรหัสห้องและรายชื่อผู้เล่นที่เข้าร่วมแล้ว โดยใช้ JavaScript Polling ความถี่ต่ำ (เช่น ทุก 15-20 วินาที) ไปยัง API เพื่อดึงรายชื่อผู้เล่นและสถานะของห้อง (ว่าครูกดเริ่มเกมหรือยัง)
* **ขั้นตอนที่ 2.3: สร้างกลไกควบคุมการเล่นใน play.php**
  + **วัตถุประสงค์:** ทำให้เกมสามารถ "หยุดรอ" คำสั่งจากครูได้
  + **การดำเนินการ:**
    1. **ฝั่งครู:** สร้างหน้า UI สำหรับควบคุมห้อง Live เช่น live\_control.php?code=A4B1 ในหน้านี้จะมีปุ่ม "ปลดล็อกด่านถัดไป", "หยุดเกมชั่วคราว", "แสดงคำใบ้"
    2. **ฝั่งนักเรียน (game\_logic.js):** เมื่อเล่นจบด่านและ Popup แสดงผลแล้ว จะเข้าสู่สถานะ "รอ"
    3. ในสถานะ "รอ" จะมีการ Polling ไปที่ API live\_status.php ทุกๆ 20 วินาที เพื่อถามว่าสถานะของ current\_stage\_id ในตาราง live\_sessions เปลี่ยนไปหรือยัง
    4. ถ้ามีการเปลี่ยนแปลง (ครูกดปลดล็อก) ปุ่ม "ไปด่านถัดไป" ถึงจะปรากฏขึ้นมาให้นักเรียนกดครับ

**เฟสที่ 3: ระบบให้รางวัลและเสริมแรง (Reward & Engagement System)**

เมื่อระบบ Live ทำงานได้แล้ว เราจะเพิ่มส่วนที่ทำให้สนุกขึ้นครับ

* **ขั้นตอนที่ 3.1: สร้าง Leaderboard แบบคะแนนรวม**
  + **วัตถุประสงค์:** แสดงอันดับผู้เล่นตามคะแนนสะสม
  + **การดำเนินการ:**
    1. สร้างหน้า leaderboard.php?session\_code=A4B1
    2. สคริปต์ในหน้านี้จะดึงข้อมูลจากตาราง progress ของผู้เล่นทุกคนที่อยู่ในเซสชั่นนั้นๆ แล้วนำ stars\_awarded มารวมกันเพื่อจัดอันดับ
    3. ครูสามารถกดปุ่ม "แสดง Leaderboard" ในหน้าควบคุมของครู เพื่อให้นักเรียนทุกคนถูกส่งมาที่หน้านี้พร้อมกัน
* **ขั้นตอนที่ 3.2: สร้างระบบแจ้งเตือนการบ้าน**
  + **วัตถุประสงค์:** แจ้งนักเรียนว่ามีงานอะไรที่ต้องทำ
  + **การดำเนินการ:**
    1. สร้างตาราง notifications (id, user\_id, message, is\_read, created\_at)
    2. **ฝั่งครู:** สร้างส่วนสำหรับส่งการแจ้งเตือนให้นักเรียน (อาจจะส่งให้ทั้งห้อง หรือรายบุคคล)
    3. **ฝั่งนักเรียน (student\_dashboard.php):** เพิ่มไอคอนกระดิ่งเล็กๆ ที่ Header เมื่อมี is\_read = 0 และเมื่อคลิกก็จะแสดงรายการแจ้งเตือน

แผนการทั้งหมดนี้จะช่วยให้เราสามารถสร้างระบบที่คุณครูต้องการได้อย่างเป็นขั้นตอน มีความเสถียร และที่สำคัญที่สุดคือ **ทำงานได้จริงบนข้อจำกัดของ Hosting ที่เรามีครับ!**