

# SIMPLE CALCULATOR

PRESENTED BY น.ส.รัฐภรณ์ ชูราศรี 1670705373  
นายจิรวัฒน์ กุลบุญ 1670706280

# OBJECTIVE

เป้าหมายหลัก:

- พัฒนา Web Application สำหรับการคำนวณทางคณิตศาสตร์และวิทยาศาสตร์
- ศึกษาและปฏิบัติจริงตามกระบวนการ SDLC (Software Development Life Cycle) ครบทั้ง 6 ขั้นตอน

สิ่งที่ได้รับ:

- เข้าใจการแปลง Requirement เป็น Software ที่ใช้งานได้จริง
- ฝึกทักษะการเขียนโค้ดด้วย HTML, CSS, JavaScript (Vanilla) และการจัดการ Algorithm

# REQUIREMENTS

## Scope (ขอบเขตงาน):

- In-Scope: บวกลบคูณหารพื้นฐาน, ฟังก์ชันตรีโกณมิติ ( $\sin$ ,  $\cos$ ,  $\tan$ ), เลขยกกำลัง ( $^$ ), สแควร์รูท ( $\sqrt{\phantom{x}}$ ), และการใช้วงเล็บ ( ) จัดลำดับความสำคัญ
- Out-of-Scope: การเก็บประวัติการคำนวณ (History), การพล็อตกราฟ

## Key User Stories:

- "ในฐานะผู้ใช้ ฉันต้องการคำนวณ  $\sin$ ,  $\cos$ ,  $\tan$  เพื่อแก้โจทย์วิทยาศาสตร์"
- "ในฐานะผู้ใช้ ฉันต้องการใส่ วงเล็บ ( ) เพื่อกำหนดลำดับการคำนวณที่ซับซ้อนได้"

# UX/UI DESIGN

Concept: Clean & Responsive Design

Design Rationale:

- Layout: ใช้ Grid Layout (5 คอลัมน์) เพื่อรองรับปุ่มฟังก์ชันวิทยาศาสตร์ที่เพิ่มเข้ามา
- Color Coding: แยกสีปุ่มชัดเจน (สีฟ้า=ตัวเลข, สีส้ม=เครื่องหมาย, สีเทา=ฟังก์ชันวิทย์) เพื่อลดความสับสนของผู้ใช้
- Feedback: มีหน้าจอแสดงผล 2 บรรทัด (Expression ด้านบน, ผลลัพธ์ด้านล่าง) ให้ผู้ใช้ตรวจสอบสิ่งที่พิมพ์ได้ง่าย

# SYSTEM DESIGN

## Architecture (C4 Model):

- Type: Client-Side Web Application (Single Page Application)
- Component: ทำงานบน Browser 100% ไม่ต้องเชื่อมต่อ Server/Database

## Sequence Diagram Logic:

- User Input → Tokenizer (แยกตัวเลข/เครื่องหมาย) → Shunting-Yard Algorithm (จัดเรียงลำดับความสำคัญ) → RPN Evaluator → Result Display

# IMPLEMENTATION

## Technology Stack:

- Frontend: HTML5, CSS3, JavaScript (ES6+)
- Tools: VS Code, Git/GitHub

## Key Algorithm:

- ใช้ Shunting-Yard Algorithm ในการแปลงนิพจน์คณิตศาสตร์ (Infix) เป็น Postfix (Reverse Polish Notation) เพื่อรองรับ Order of Operations (PEMDAS)



**THANK  
YOU**