

# PORT FOLIO

แฟ้มสะสมผลงาน



MUHAMMAD-AMIN KAMASEH

**มูฮัมหมัดอามีน กามาเซะ**

Prince of Songkla University, Pattani Campus

คณะวิทยาการสื่อสาร

สาขาคอมพิวเตอร์และวิทยาการสารสนเทศเพื่อการจัดการ

2564 - 2567

**3.32/4.0 GPAX**

# คำนำ

สวัสดีครับ

ผมชื่อ มุฮัมหมัดอามีน กามาเซะ เป็นนักศึกษาจบใหม่จากคณะวิทยาการสื่อสาร สาขาคอมพิวเตอร์และการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี

ผมมีความสนใจและหลงใหลในการพัฒนาเว็บไซต์ และมีความตั้งใจที่จะเรียนรู้และพัฒนาทักษะด้านการเขียนโปรแกรมอย่างต่อเนื่อง ด้วยความรู้ความสามารถที่สั่งสมมาทั้งจากการเรียนในมหาวิทยาลัย การเข้าร่วมกิจกรรมชมรม IT และการฝึกงานในตำแหน่ง Software Developer ผมได้ฝึกฝนการทำงานจริง การแก้ปัญหา และการทำงานร่วมกับผู้อื่นในรูปแบบทีม

Portfolio เล่มนี้ รวบรวมประสบการณ์ ทักษะ และผลงานที่ผมภาคภูมิใจ หวังเป็นอย่างยิ่งว่าจะเป็นส่วนหนึ่งที่สะท้อนความสามารถและความตั้งใจของผมในการเริ่มต้นเส้นทางสายอาชีพในสายงานเทคโนโลยี

ขอบคุณที่สละเวลาเปิดอ่านครับ

จัดทำโดย

มุฮัมหมัดอามีน กามาเซะ



# ประสพการณ์

# ฝึกสหกิจศึกษา

## Software Developer Intern

Mango Consultant Co. Ltd | 2024-2025 (4 Months)



### การทำงาน

- ทำงานร่วมกับทีมไอทีในการแก้ไขข้อผิดพลาดของระบบและปรับปรุงประสิทธิภาพ
- ได้รับประสบการณ์ตรงในการแก้ไขปัญหาทางเทคนิค
- ใช้ Git ในการควบคุมเวอร์ชันและนำแนวทาง Agile มาใช้

**พลาซ่า**

# ผลงาน

## โปรเจกต์ “ระบบประมูลสินค้า (Fast Bidding)”

### การทำงาน

#### 1. ผู้ใช้งานทั่วไป (User)

ลงทะเบียน → เข้าสู่ระบบ → เลือกสินค้า → เข้าร่วมประมูล → หากชนะ → ดำเนินการจ่ายเงิน → รอรับสินค้า

#### 2. ผู้ใช้งานที่เปิดร้าน (Store Owner)

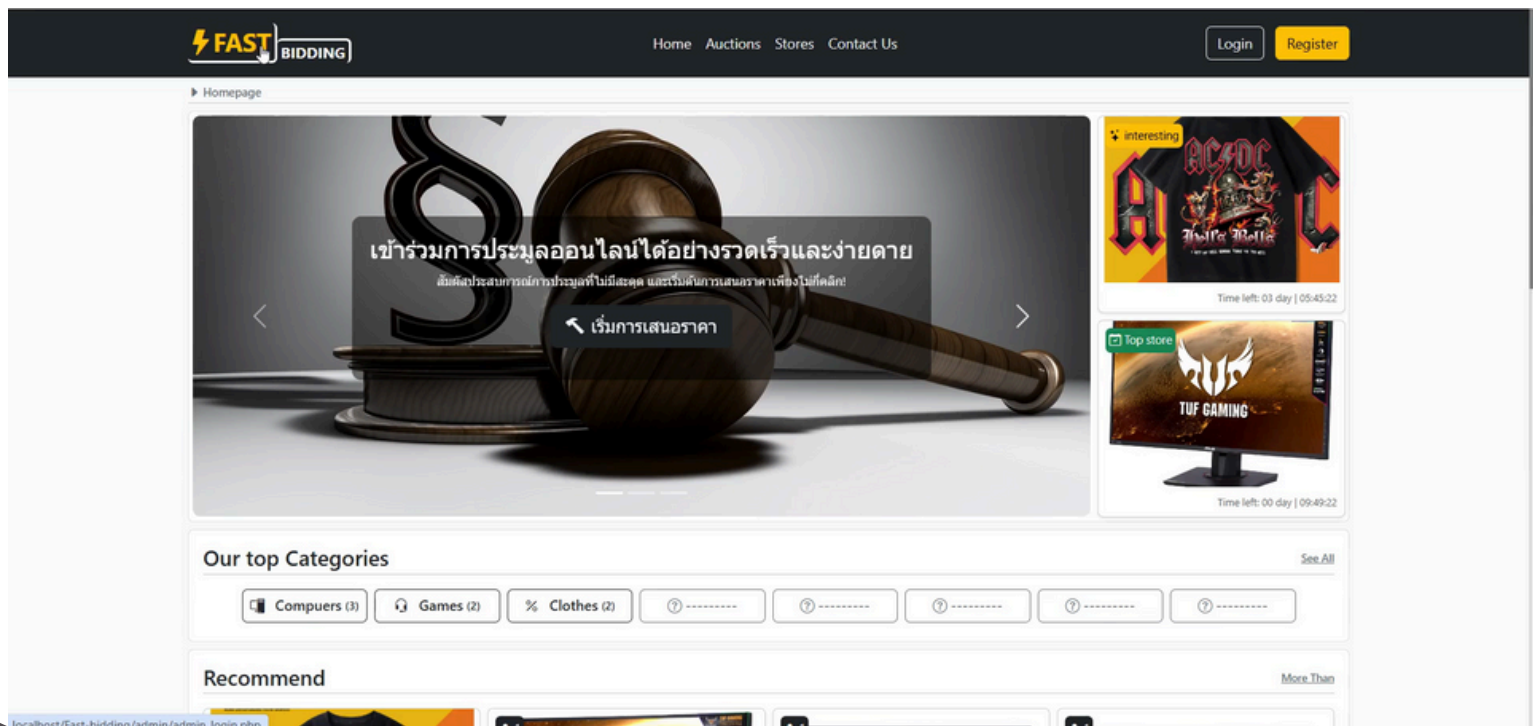
ลงทะเบียนเปิดร้าน → สถานะเปลี่ยนเป็นเจ้าของร้าน → เพิ่มสินค้าลงระบบ → ดูสถานะการประมูล → หากสินค้าขายได้ → รอให้ระบบโอนเงินหลังหัก 10% → สามารถกด “ถอนเงิน”

#### 3. ผู้ดูแลระบบ (Admin)

ตรวจสอบคำขอเปิดร้าน → ตรวจสอบการชำระเงินของผู้ชนะ → ยืนยันการถอนเงินให้เจ้าของร้าน

### การแสดงผล UI

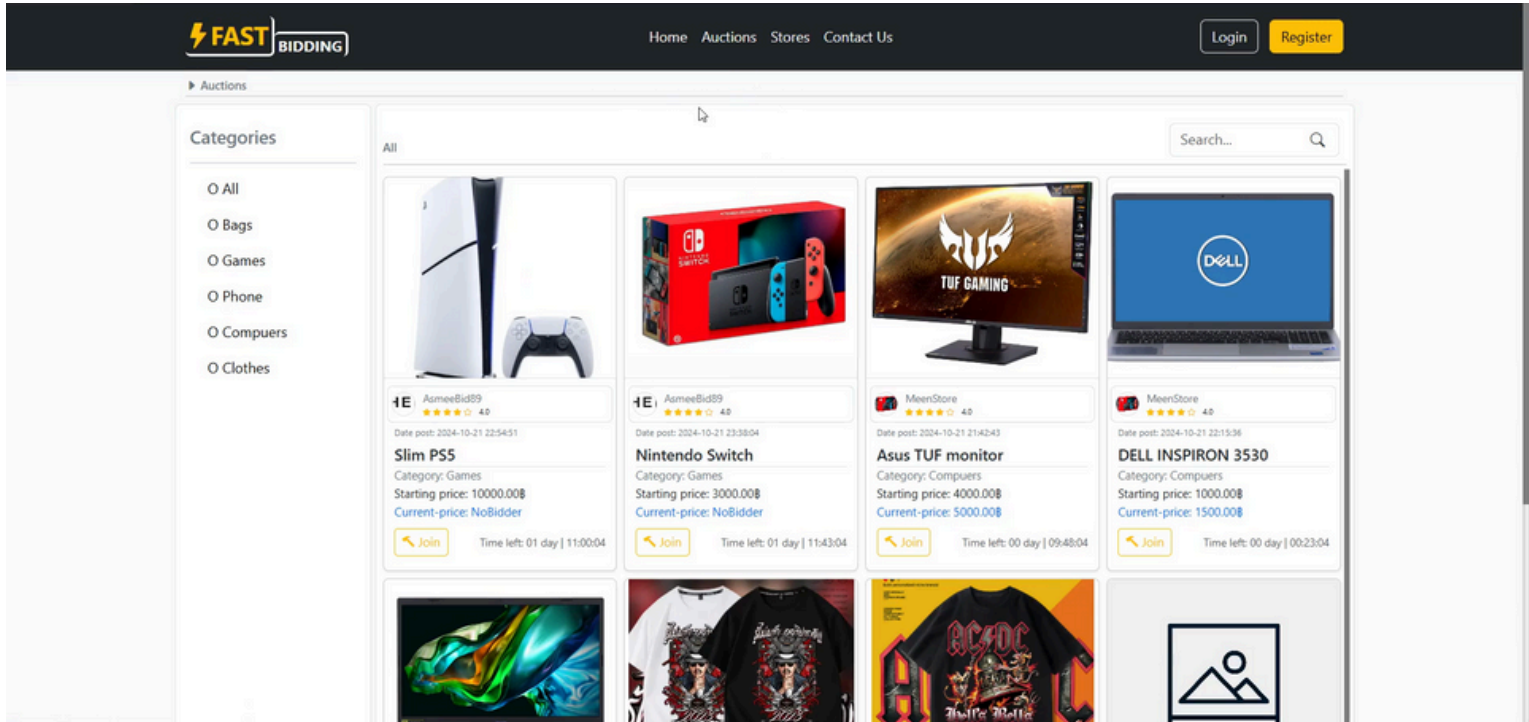
- หน้า HOMEPAGE



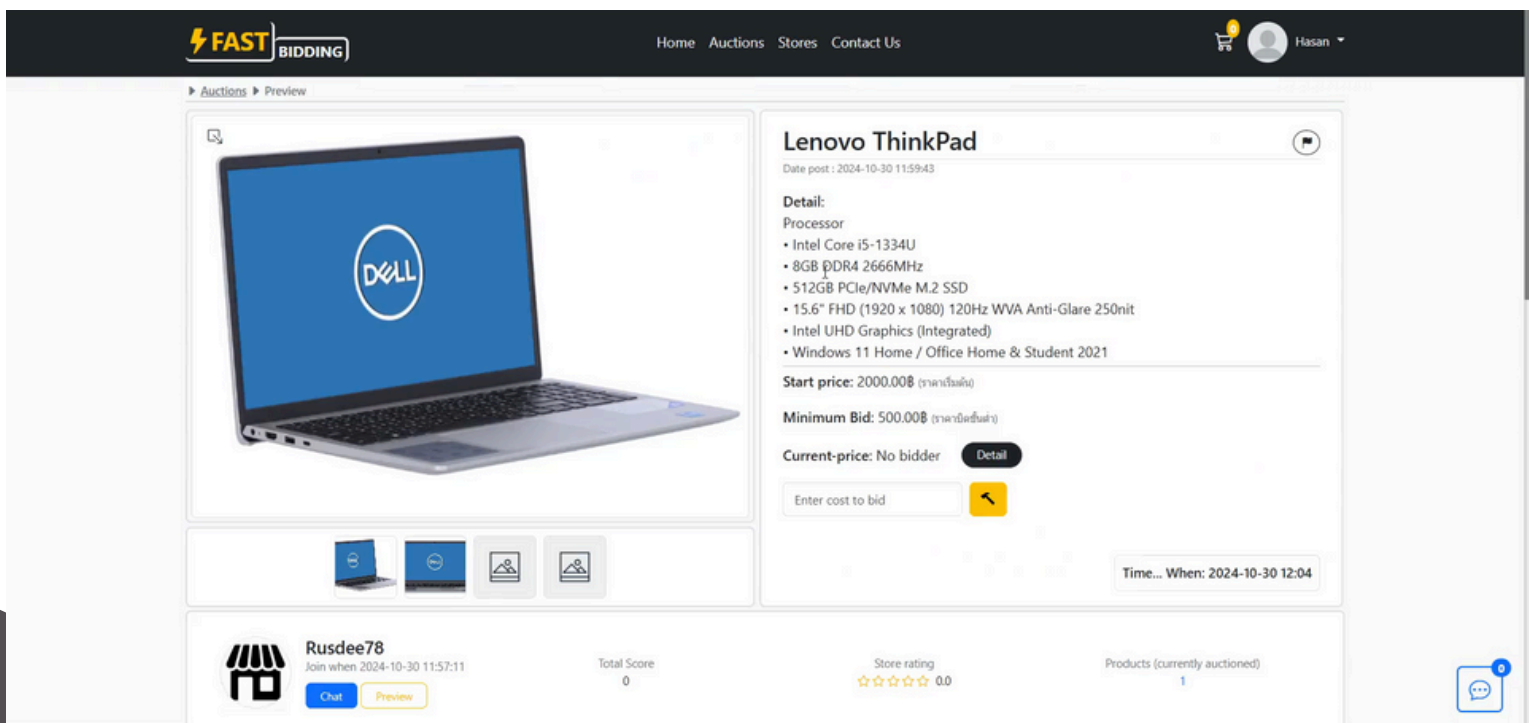
# โปรเจกจบ “ระบบประมูลสินค้า” (ต่อ)

## การแสดงผล UI

### • หน้า AUCTION



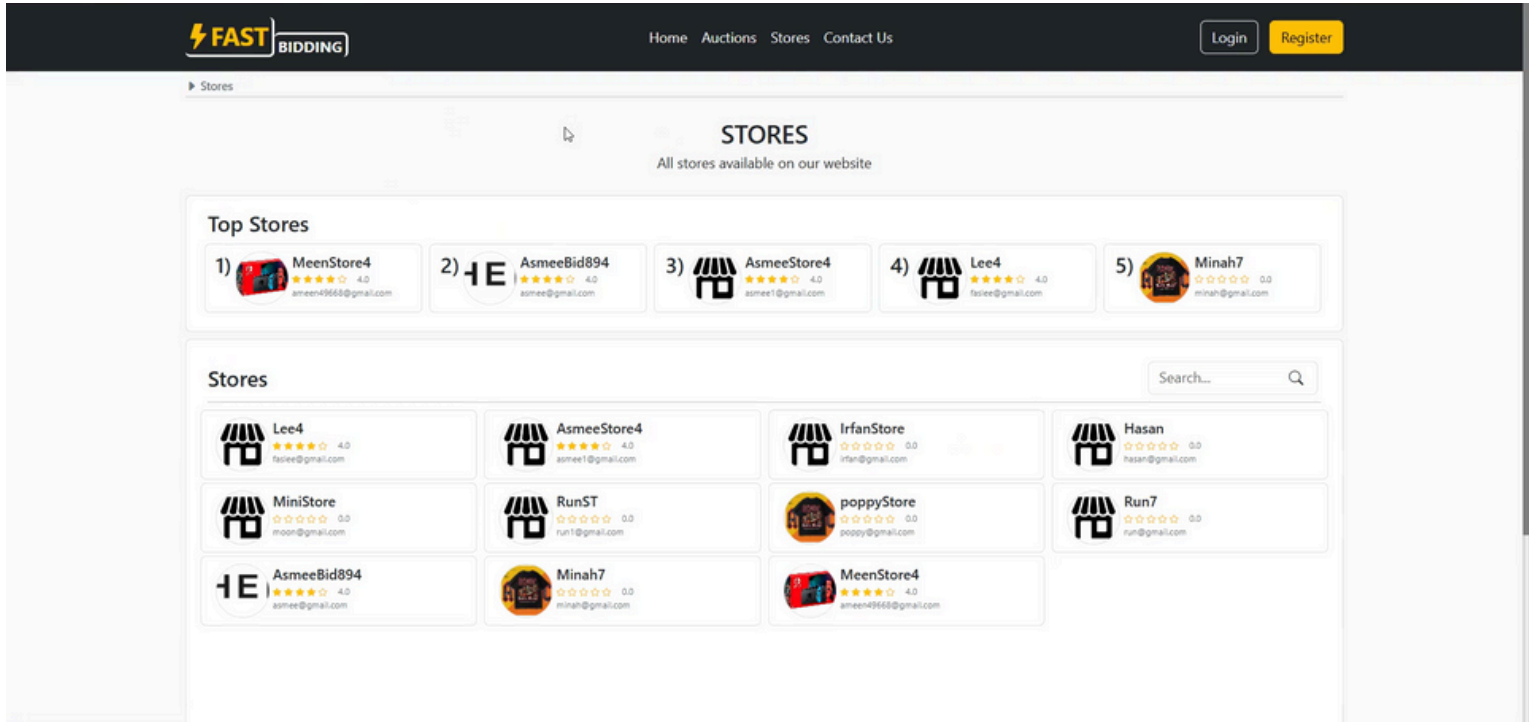
### • หน้า PREVIEW



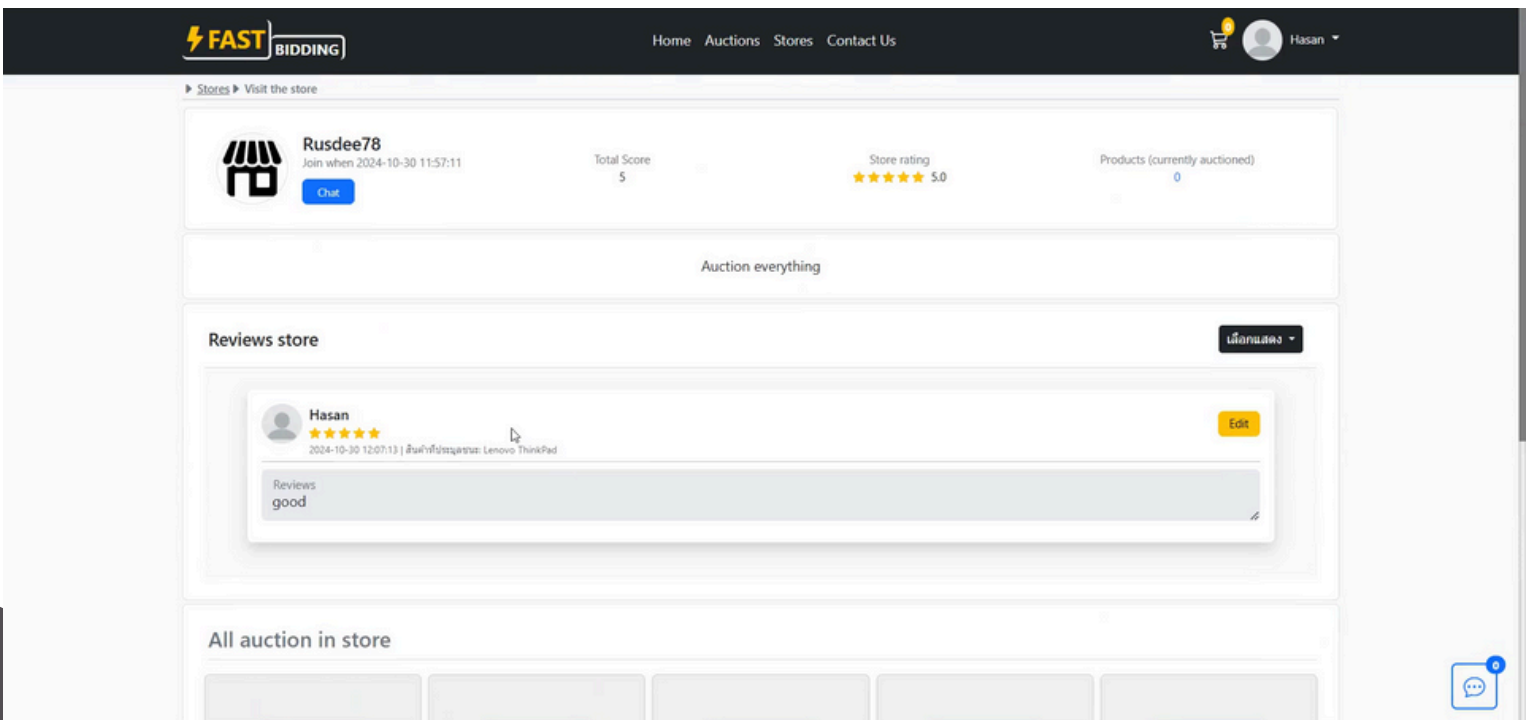
# ไปเจอกับ “ระบบประมูลสินค้า” (ต่อ)

## การแสดงผล UI

- หน้า STORE



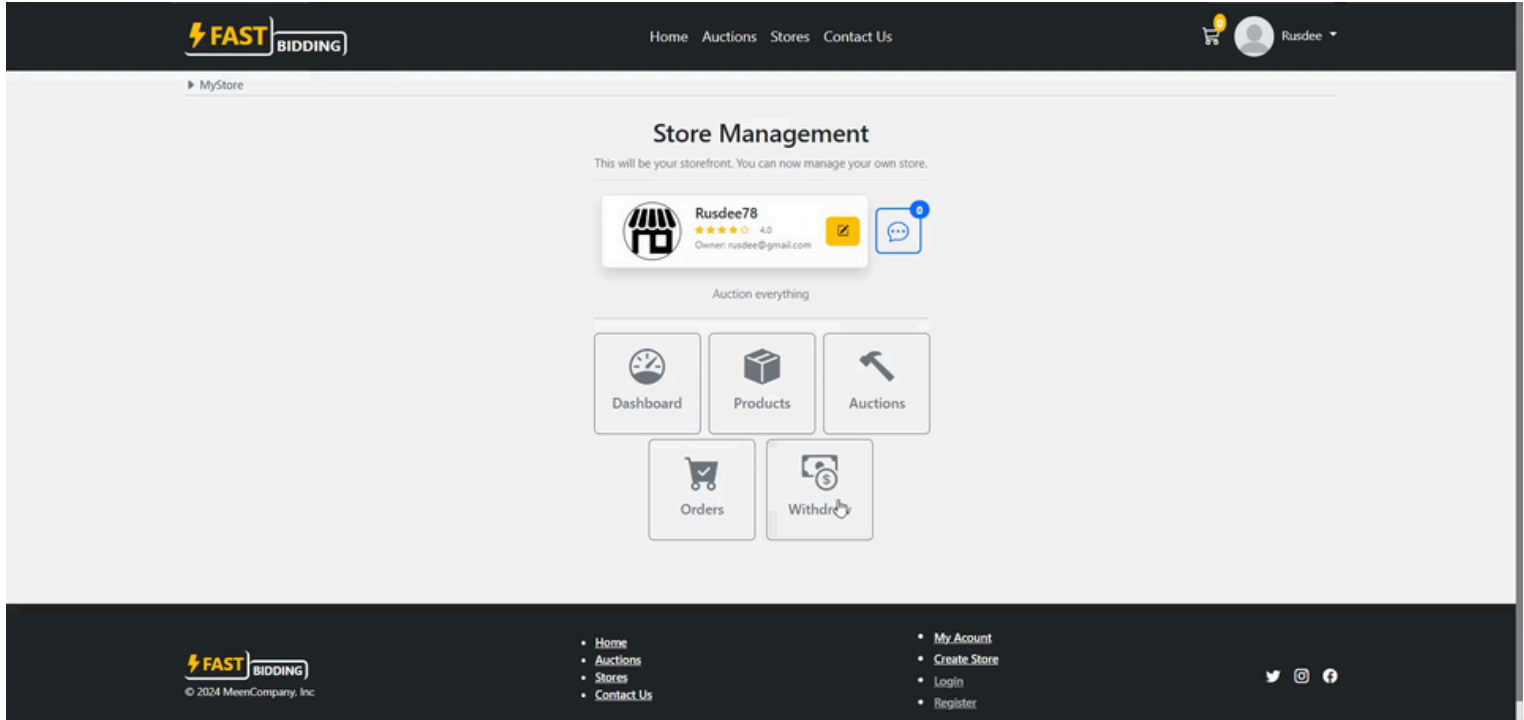
- หน้า VISIT THE STORE



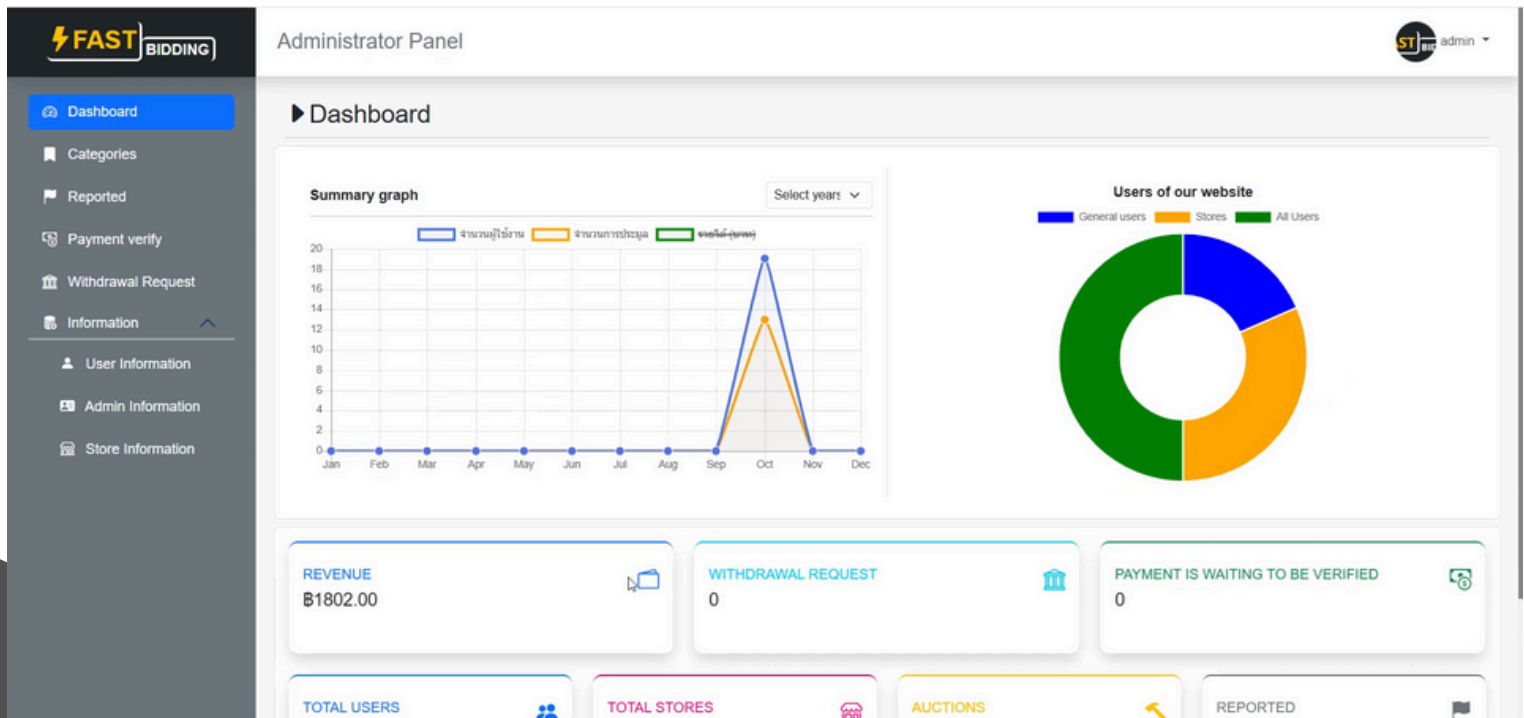
# โปรเจกจบ “ระบบประมูลสินค้า” (ต่อ)

## การแสดงผล UI

- หน้า STORE MANAGEMENT

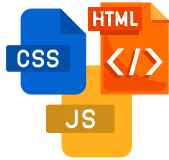


- หน้า DASHBOARD



# โปรเจกจบ “ระบบประมูลสินค้า” (ต่อ)

## เครื่องมือและภาษาโปรแกรม



## ตัวอย่างโค้ด

- เสนอราคาการประมูล หรือเข้าร่วมการประมูล

```
<?php
include('../function.php');
$db = new DB_con();
$user = $_POST['user'];
$auc_id = $_POST['auc_id'];
$price = $_POST['price'];
/*$db->send_bid($user, $auc_id, $price);*/

date_default_timezone_set('Asia/Bangkok');
$currentDateTime = date('Y-m-d H:i:s');

$result = $db->show_table('auctions as a LEFT JOIN biddings as b ON a.AuctionID = b.AuctionID WHERE a.Time_end > NOW() AND a.AuctionID = '.$auc_id.' ORDER BY b.Bid_price DESC LIMIT 1');
foreach($result as $re) {
    $AuctionID = $re['AuctionID'];
    $start_price = $re['Start_price'];
    $mini_bid = $re['Mini_bid'];
    $current_price = $re['Bid_price'];
    $StoreID = $re['StoreID'];
    $Time_end = $re['Time_end'];
}

$check_status = $db->show_table('auctions as a, stores as s WHERE a.StoreID=s.StoreID AND a.AuctionID = '.$auc_id);
foreach($check_status as $check_st) {
    $UserID = $check_st['UserID'];
}

if($_SESSION['UserID'] != $UserID) {
    if($Time_end < $currentDateTime) {
        if($price >= $start_price+$mini_bid && $price >= $mini_bid+$current_price && $price >= $current_price) {
            $db->send_bid($user, $auc_id, $price);
            echo json_encode(["success" => true]);
        } else {
            if(isset($current_price)) {
                $start_price = 0;
            }
            $_SESSION['Sweet_Alert'] = array('error', 'กรุณาสอบราคาที่เหมาะสม! (ขั้นต่ำ: '.$mini_bid+$current_price+$start_price.'฿', 2500);
            echo json_encode(["success" => false, "redirect" => "/fast-bidding/auction/preview.php?AID=".$auc_id]);
        }
    } else {
        $_SESSION['Sweet_Alert'] = array('error', 'ประมูลไม่ได้ เนื่องจากประมูลได้ถึงแล้ว', 2500);
        echo json_encode(["success" => false, "redirect" => "/fast-bidding/auction/preview.php?AID=".$auc_id]);
    }
} else {
    $_SESSION['Sweet_Alert'] = array('error', 'เข้าร่วมประมูลไม่ได้ เนื่องจากประมูลถึงจุดจำกัดแล้ว', 2500);
    echo json_encode(["success" => false, "redirect" => "/fast-bidding/auction/preview.php?AID=".$auc_id]);
}
}
```

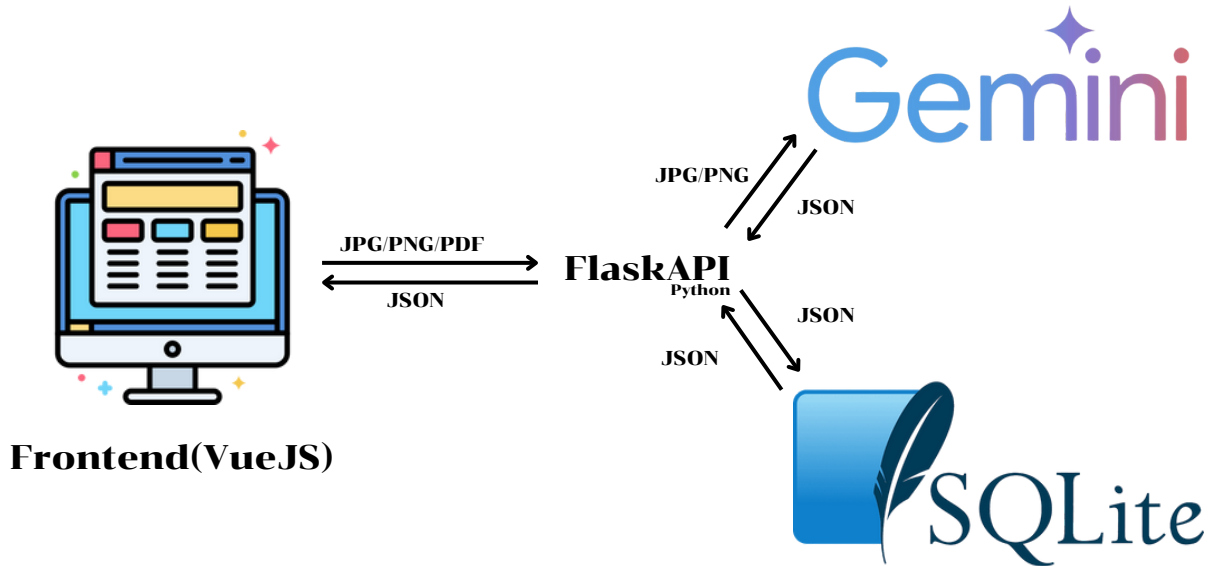
Video ตัวอย่างการทำงาน :

<https://drive.google.com/file/d/12I3jjOaRSUQXRpQilQbwO5KeoOtgaude/view>

# ผลงาน

## โปรเจกต์หลัก "OCR Extract Receipt"

### แผนภาพโครงสร้างระบบ

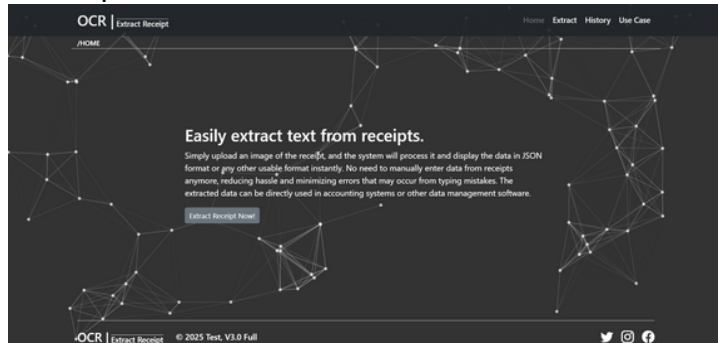


### การทำงาน

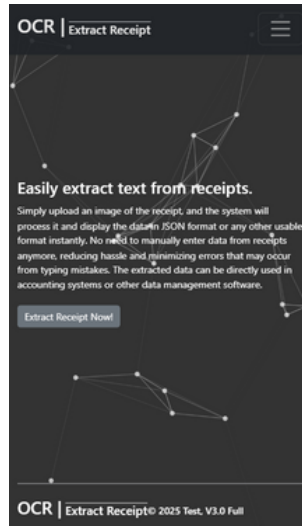
- ผู้ใช้อัปโหลดไฟล์ (JPG/PNG หรือ PDF) ผ่าน Vue.js หรือหน้าบ้าน
- Flask API ส่งไฟล์ไปยัง Gemini เพื่อวิเคราะห์ และบันทึกข้อมูลลง SQLite
  - ถ้าเป็นไฟล์ PDF ก็จะแปลงเป็น JPG ก่อนแล้วค่อยส่งให้ Gemini
- Vue.js รับข้อมูลที่ประมวลผลแล้วและแสดงผลให้ผู้ใช้

### การแสดงผล UI

#### Desktop



#### Mobile



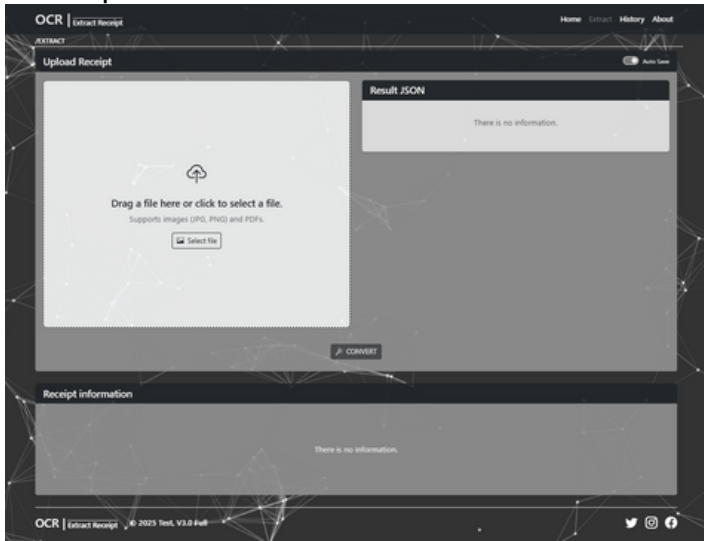
#### IPAD



# โปรเจกต์ “OCR Extract Receipt” (ต่อ)

## การแสดงผล UI

### Desktop



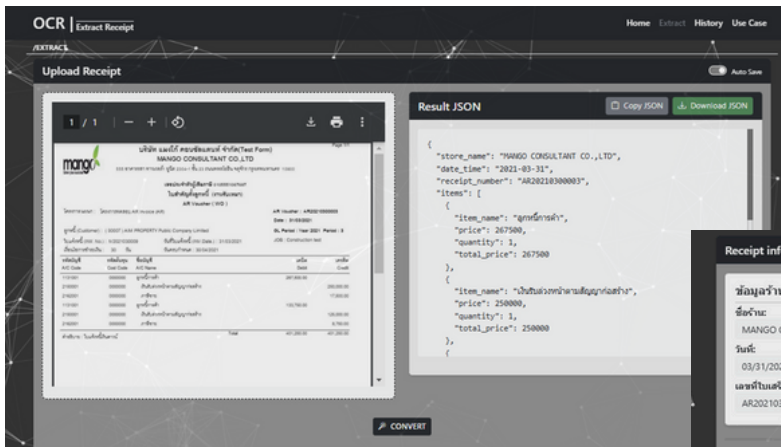
### Mobile



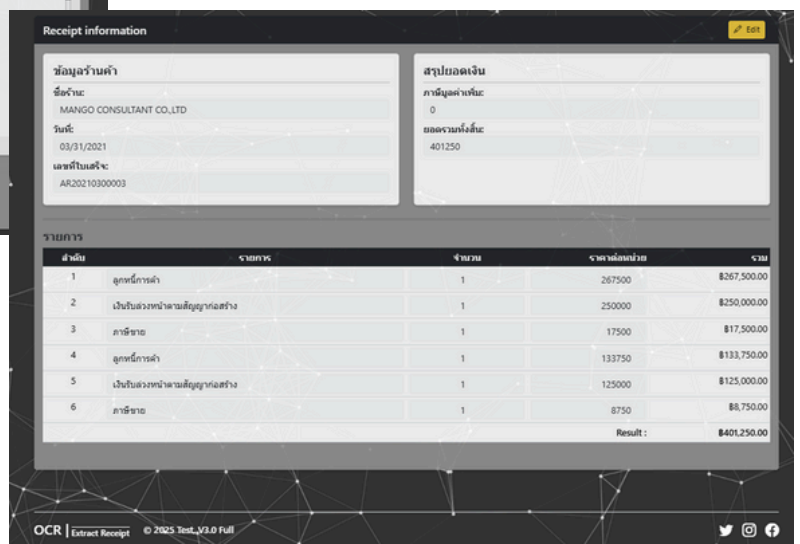
### IPAD



## ผลลัพธ์หลังประมวลผล

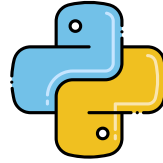


- ผลลัพธ์ที่ได้



# โปรเจกต์สกริปต์ “OCR Extract Receipt” (ต่อ)

## เครื่องมือและภาษาโปรแกรม



## ตัวอย่างโค้ด

```

if file_ext == ".pdf":
    try:
        file.save(file_path) # บันทึกไฟล์ PDF ชั่วคราว
        jpg_paths = convert_pdf_to_jpg(file_path) # แปลง PDF เป็น JPG
        os.remove(file_path) # ลบไฟล์ PDF ชั่วคราวทิ้งแล้ว

        if not jpg_paths:
            return jsonify({"error": "Failed to convert PDF to images"}), 500

        file_path = jpg_paths[0] # ได้รูปถ่ายจากขั้นตอนการแปลง PDF
    except Exception as e:
        return jsonify({"error": f"PDF conversion failed: {str(e)}"}), 500
    else:
        file.save(file_path) # บันทึกภาพที่ได้รูปถ่าย

    try:
        sample_file = prep_image(file_path)
        prompt = ("extract the text in the image verbatim and identify: store name, Date(format: yyyy-mm-dd), Receipt  

        text = extract_text_from_image(sample_file, prompt)

        if text:
            structured_data = parse_extracted_text(text)
            return jsonify(structured_data), 200
        else:
            return jsonify({"error": "Failed to extract text from the image."}), 500
    except Exception as e:
        return jsonify({"error": str(e)}), 500
    
```

ตัวแปรของโมเดล	ต้นทุน	เวลาที่ใช้	เพิ่มประสิทธิภาพสำหรับ
Gemini 2.0 Flash gemini-2.0-flash	เสียง รูปภาพ วิดีโอ และข้อความ	ข้อความ รูปภาพ (เร้า นี) และ เสียง (เร้า นี)	เพิ่มประสิทธิภาพและลดการสร้างแบบจำลองดีเลย์ลงครึ่งหนึ่งสำหรับงานที่มีความหลากหลาย
Gemini 2.0 Flash-Lite gemini-2.0-flash-lite	เสียง รูปภาพ วิดีโอ และข้อความ	ข้อความ	โมเดล Gemini 2.0 Flash ที่เพิ่มประสิทธิภาพเพื่อประหยัดต้นทุนและเวลาในการตอบสนอง
Gemini 1.5 Flash gemini-1.5-flash	เสียง รูปภาพ วิดีโอ และข้อความ	ข้อความ	ประสิทธิภาพที่รวดเร็วและหลากหลายสำหรับงานต่างๆ
Gemini 1.5 Flash-8B gemini-1.5-flash-8b	เสียง รูปภาพ วิดีโอ และข้อความ	ข้อความ	งานจำนวนมากและงานที่มีระดับปัญหาต่ำ
Gemini 1.5 Pro gemini-1.5-pro	เสียง รูปภาพ วิดีโอ และข้อความ	ข้อความ	งานที่มีเหตุผลที่ซับซ้อนซึ่งต้องใช้ความสามารถมากขึ้น
การฝังข้อความ text-embedding-004	ข้อความ	ข้อความที่ฝัง	การวัดความเกี่ยวข้องของข้อความ

```

# ฟังก์ชันดึงข้อความจากรูปภาพ
def extract_text_from_image(image_path, prompt):
    model = genai.GenerativeModel(model_name="gemini-1.5-pro")
    response = model.generate_content([image_path, prompt])
    return response.text
    
```

Video ตัวอย่างการทำงาน :

<https://drive.google.com/file/d/12I3jjOaRSUQXRpQilQbwO5KeoOtgaude/view>

The background features abstract geometric shapes in red and grey. In the top-left corner, there is a grey square partially overlapping a red triangle. In the bottom-right corner, there is a grey trapezoid partially overlapping a red triangle. A thin red line runs diagonally across the lower half of the page.

**Thanks**