



โครงการ

Marine Animals Voting System (ระบบโหวตสัตว์ทะเลที่น่ารักที่สุด)

จัดทำโดย

6704062611328 ณพพัฒน์ ลัทธิ

เสนอ

ผู้ช่วยศาสตราจารย์ ดร. สติติ์ ประสมพันธ์

วิชา 040613204 Object-Oriented Programming

ภาคเรียนที่ 1/2567

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เกี่ยวกับโครงการ

ชื่อโปรเจก : Marine Animals Voting System (ระบบโหวตสัตว์ทะเลที่น่ารักที่สุด)

นำเสนอโดย : นาย ณพัฒน์ ลัทธิ

อาจารย์ผู้สอน : ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

Source Code : <https://github.com/Napat199/SourceCodeMarineSystem>

WebSite : <https://marine.kesug.com/?i=1>

บทที่ 1 ที่มาและความสำคัญของโครงการ

โครงการนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object-Oriented Programming โดยการนำเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบเว็บแอปพลิเคชัน โดยใช้แนวคิดการเขียนโปรแกรมแบบเชิงวัตถุทั้งในส่วน Backend (PHP) และ Frontend (JavaScript) โครงการนี้เป็นระบบโหวตสัตว์ทะเลที่ผู้ใช้สามารถโหวตสัตว์ทะเลที่ชอบ ดูตารางคะแนน และสถิติการโหวตแบบ Real-time พร้อมทั้งมี GUI ที่สวยงามพร้อม Event Handling และ Database Integration ครบถ้วน

ประเภทของโครงการ

โปรแกรมเว็บแอปพลิเคชัน Full-Stack

Frontend: HTML5, CSS3, JavaScript (OOP)

Backend: PHP (OOP)

Database: MySQL

ประโยชน์

เพื่อนำหลักการ OOP มาประยุกต์ใช้จริง ทั้ง Inheritance, Polymorphism, Interface, Abstract Class

เพื่อพัฒนาทักษะการเขียนโปรแกรม Full-Stack ตั้งแต่ Frontend จนถึง Backend และ Database

เพื่อเรียนรู้การจัดการ Database โดยใช้ MySQL ในการเก็บข้อมูลสัตว์และคะแนนโหวต

เพื่อพัฒนา GUI และ Event Handling ที่ใช้งานง่ายและสวยงาม

เพื่อเป็นแหล่งข้อมูลสัตว์ทะเล พร้อมระบบโหวตแบบ Interactive

ขอบเขตของโครงการ

1. ความต้องการของระบบ (Functional Requirements)

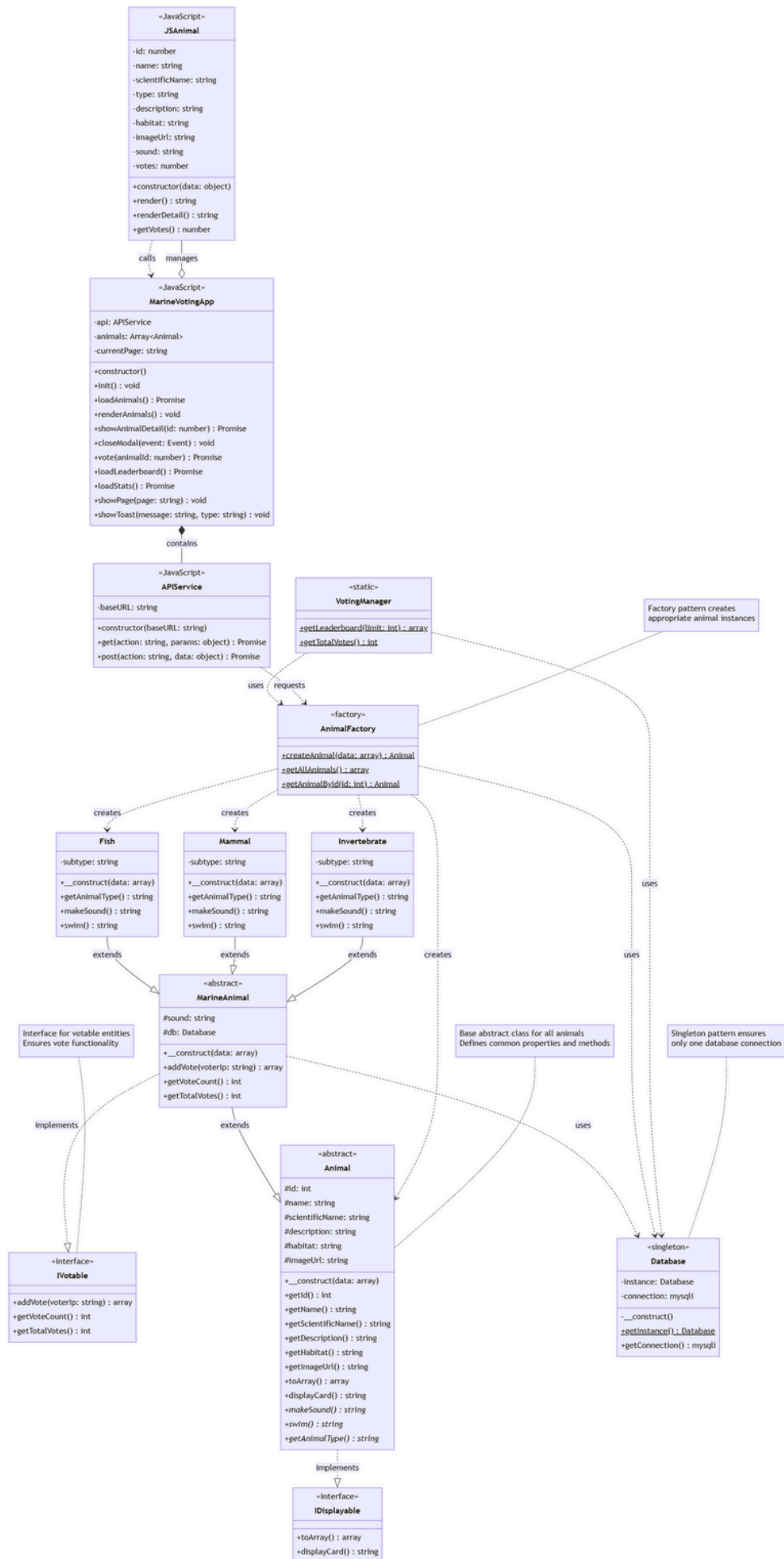
- ระบบแสดงสัตว์ทะเล: แสดงข้อมูลสัตว์ทะเลพร้อมรูปภาพและรายละเอียด
- ระบบโหวต: ผู้ใช้สามารถโหวตสัตว์ที่ชอบได้ (จำกัด 1 ครั้ง/วัน/IP)
- ตารางคะแนน (Leaderboard): แสดงอันดับสัตว์ที่ได้รับความนิยม
- สถิติการโหวต: แสดงสถิติการโหวตรายวัน และจำนวนสัตว์แต่ละประเภท
- Responsive Design: รองรับทุกขนาดหน้าจอ

2. ตารางเวลาการดำเนินโครงการ (Project Schedule)

งาน	สัปดาห์ 1	สัปดาห์ 2	สัปดาห์ 3	สัปดาห์ 4	สัปดาห์ 5
1. ออกแบบและสร้าง Database	✓				
2. สร้าง PHP OOP Classes (Animal, Fish, Mammal, etc.)	✓	✓			
3. สร้าง API Endpoints		✓	✓		
4. สร้าง Frontend HTML/CSS			✓	✓	
5. สร้าง JavaScript OOP Application			✓	✓	
6. ระบบโหวตและ Database Integration				✓	✓
7. Leaderboard และ Statistics				✓	✓
8. GUI Enhancement (Animations, Effects)					✓
9. Testing และ Bug Fixes					✓
ความคืบหน้า	10%	25%	45%	70%	100%

บทที่ 2 การพัฒนา

แผนภาพ Class Diagram



โครงการนี้มีทั้งหมด 13 Class (HTML,CSS,JAVASCRIPT,PHP,DATABASE)

IVotable Interface : ทำให้สัตว์ทุกตัวสามารถรับโหวตได้

IDisplayable Interface : ทำให้สัตว์ทุกตัวแสดงผลในรูปแบบเดียวกัน

Animal (Abstract Class) : คลาสพื้นฐานของสัตว์ทั้งหมด (พ่อแม่ใหญ่)

MarineAnimal (Abstract Class) : มีระบบโหวต (implements IVotable), เชื่อมต่อ Database

Fish Class : คลาสสำหรับปลาทุกชนิด

Mammal Class : คลาสสำหรับสัตว์เลี้ยงลูกด้วยนมในทะเล

Invertebrate Class : สัตว์ไม่มีกระดูกสันหลัง (กลุ่มปลาหมึก แมงกะพรุน)

Database Class : คลาสสำหรับเชื่อมต่อ Database

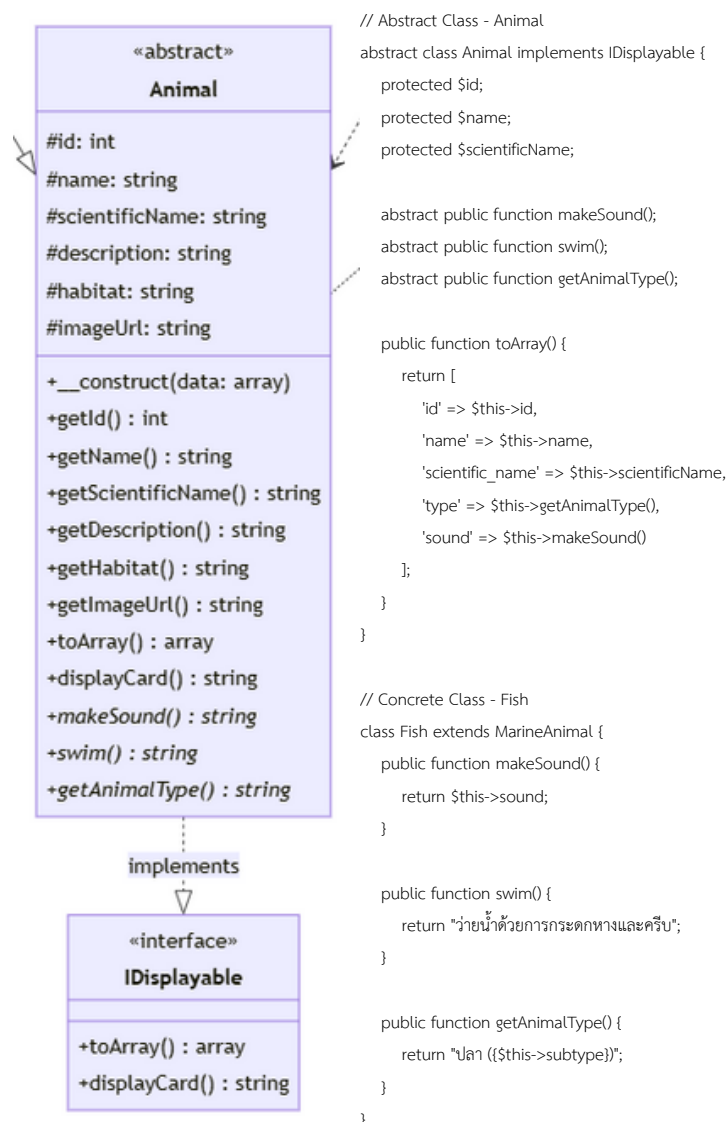
AnimalFactory Class : คลาสสำหรับสร้าง Animal Objects

VotingManager Class : คลาสจัดการสถิติการโหวต

Animal Class (JavaScript) : คลาส JavaScript สำหรับแสดงข้อมูลสัตว์

APIService Class : คลาสจัดการการเรียก API

MarineVotingApp Class : คลาสหลักของ Website, โหลดข้อมูลจาก API, แสดงผลบนหน้าเว็บ, จัดการการโหวต, แสดงLeaderboard และ Stats



ตัวอย่างการ Implement Abstract Class - Animal

รูปแบบการพัฒนาโครงการ

Backend (PHP)

ภาษา: PHP 7.4+

Database: MySQL 8.0

ORM Pattern: Active Record Pattern

Design Patterns:

Singleton (Database)

Factory (AnimalFactory)

Strategy (Different animal types)

API Style: RESTful API

Frontend (JavaScript)

ภาษา: JavaScript ES6+ (OOP)

CSS Framework: Tailwind CSS

Design Pattern: MVC-like pattern

HTTP Client: Fetch API

Database

ชื่อฐานข้อมูล: marine_voting_system

animals: เก็บข้อมูลสัตว์ทะเล

votes: เก็บคะแนนโหวต

users: เก็บข้อมูลผู้ใช้ (optional)

Tools และ Libraries

XAMPP: Apache + MySQL + PHP

Tailwind CSS: Utility-first CSS framework



แนวความคิดเขียนโปรแกรมเชิงวัตถุ

Constructor

```
class Animal {  
  constructor(data) {  
    this.id = data.id;  
    this.name = data.name;  
    this.scientificName = data.scientific_name;  
    this.type = data.type;  
    this.description = data.description;  
    this.habitat = data.habitat;  
    this.imageUrl = data.image_url;  
    this.sound = data.sound;  
    this.votes = data.votes || 0;  
  }  
}
```

สร้าง Constructor เพื่อรับข้อมูล

Inheritance & Abstract

```
abstract class MarineAnimal extends Animal implements IVotable {  
  ...  
  // Implementation ของ IVotable  
  public function addVote($voterIp) {  
    $conn = Database::getInstance()->getConnection();  
    $date = date('Y-m-d');  
  
    // ตรวจสอบว่าโหวตแล้วหรือยัง (จำกัด 1 ครั้ง/วัน)  
    $stmt = $conn->prepare("  
      SELECT COUNT(*) as count FROM votes  
      WHERE animal_id = ? AND voter_ip = ? AND vote_date = ?  
    ");  
    $stmt->bind_param("iss", $this->id, $voterIp, $date);  
    $stmt->execute();  
    $result = $stmt->get_result()->fetch_assoc();  
  
    if ($result['count'] > 0) {  
      return ['success' => false, 'message' => 'คุณโหวตสัตว์นี้วันนี้แล้ว'];  
    }  
  
    ...  
  }
```

มีการสืบทอด Animal
กำหนด Contract ที่ subclass ต้องทำตาม

Polymorphism

<pre>class Fish extends MarineAnimal { private \$subtype; public function __construct(\$data = []) { parent::__construct(\$data); \$this->subtype = \$data['subtype'] ?? 'General Fish'; } public function getAnimalType() { return "ปลา ({ \$this->subtype })"; } public function makeSound() { return \$this->sound; } public function swim() { return "ว่ายน้ำด้วยการกระดกหางและครีบ"; } }</pre>	แต่ละคลาสมีการทำงานของเมธอดเดียวกันแต่ต่างกัน (Override)
--	---

Encapsulation

<pre>class Animal { protected \$id; // Protected - เข้าถึงได้จาก subclass protected \$name; private \$internalData; // Private - เข้าถึงได้เฉพาะ class นี้ public function getId() { // Public getter return \$this->id; } protected function validateData() { // Protected method // Validation logic } }</pre>	ควบคุมการเข้าถึงผ่าน getter/setter, ป้องกันการเข้าถึงข้อมูลโดยตรง
--	---

Composition

<pre>class MarineVotingApp { constructor() { this.api = new APIService('api.php'); // Composition this.animals = []; } }</pre>	มีการเรียก API
--	----------------

Algorithm

```
public function addVote($voterIp) {
    // 1. ตรวจสอบว่าเคยโหวตวันนี้แล้วหรือยัง
    $date = date('Y-m-d');
    $stmt = $conn->prepare("
        SELECT COUNT(*) FROM votes
        WHERE animal_id = ? AND voter_ip = ? AND vote_date = ?
    ");

    // 2. ถ้ายังไม่โหวต ให้เพิ่มคะแนน
    if ($count == 0) {
        $stmt = $conn->prepare("
            INSERT INTO votes (animal_id, voter_ip, vote_date)
            VALUES (?, ?, ?)
        ");
        return ['success' => true];
    }

    // 3. ถ้าโหวตแล้ว ให้แจ้งเตือน
    return ['success' => false, 'message' => 'คุณโหวตวันนี้แล้ว'];
}
```

Algorithm สำหรับการโหวต

```
public static function getLeaderboard($limit = 10) {
    $query = "
        SELECT a.*, COUNT(v.id) as vote_count
        FROM animals a
        LEFT JOIN votes v ON a.id = v.animal_id
        GROUP BY a.id
        ORDER BY vote_count DESC
        LIMIT ?
    ";

    // สร้าง Animal objects พร้อมคะแนนโหวต
    while ($row = $result->fetch_assoc()) {
        $animal = AnimalFactory::createAnimal($row);
        $leaderboard[] = [
            'animal' => $animal,
            'votes' => $row['vote_count']
        ];
    }

    return $leaderboard;
}
```

อัลกอริทึมการคำนวณ Leaderboard

บทที่ 3 สรุป

ปัญหาที่พบระหว่างการพัฒนา

ถ้าใช้โทรศัพท์เข้าเว็บจะดูบ๊คๆ Scale เพี้ยน

ตอนเขียน เชื่อม API เข้ากับ DATA BASE ค่อนข้างง

จุดเด่นของโปรแกรม

UI/UX ที่สวยงามและใช้งานง่าย

Leaderboard แบบ Real-time

