

SportHub: ระบบจัดการสนามกีฬาออนไลน์แบบครบวงจร

(SportHub: Online Sport Facility Management System)

กลุ่มเรียนที่ 2 กลุ่มที่ 3

โครงงานระยะที่ 2

โดย

นาย ศศพร ฉิมไทย 6787032

นาย นันธนันท์ ประดับมุข 6787047

นาย พุฒิสรรค์ ขมีนเครือ 6787066

นาย สรวิชญ์ ใจยินดี 6787081

รายงานนี้เป็นส่วนหนึ่งของการศึกษาเพื่อ

ให้เป็นไปตามข้อกำหนดของรายวิชา

ITDS241 Web Technologies and Applications

คณะเทคโนโลยีสารสนเทศและการสื่อสาร

มหาวิทยาลัยมหิดล

2568

สารบัญ

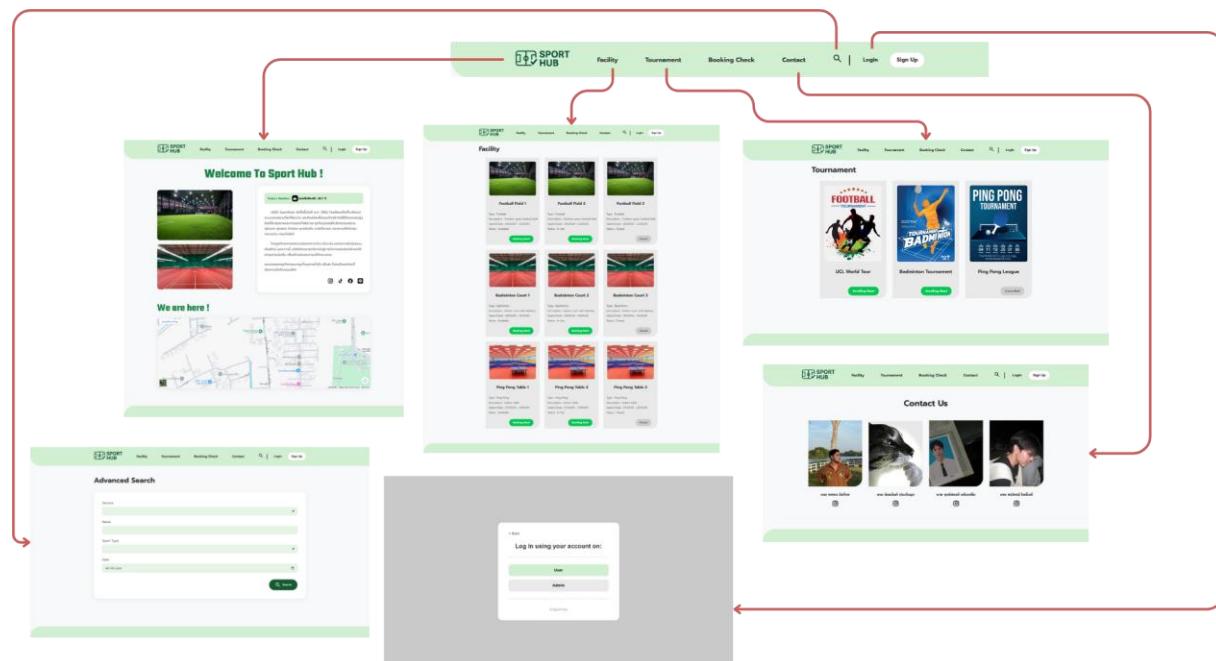
หัวข้อ	หน้า
ข้อมูลองค์รวมของโครงการ	1
แผนผังหน้าต่างๆ ของเว็บแอปพลิเคชัน	1
รายละเอียดหน้าเว็บต่างๆ	6
รายละเอียดของเว็บเซอร์วิส และโค้ด	16
ผลการทดสอบของเว็บเซอร์วิสทั้งหมด โดยใช้โปรแกรม Postman	39
ข้อความเพิ่มเติมสำหรับการใช้เทคโนโลยีปัญญาประดิษฐ์	50
แหล่งอ้างอิง	51
ภาคผนวก	52

ข้อมูลองค์รวมของโครงการ

บริษัท SportHub ก่อตั้งขึ้นเพื่อเป็นศูนย์กลางการบริการด้านการกีฬาแบบครบวงจร โดยมีสานมกีฬา หลากหลายประเภท เช่น ฟุตบอล บาสเกตบอล แบดมินตัน และปิงปอง ซึ่งให้บริการแก่ทั้งผู้เล่นทั่วไป นักเรียน นักศึกษา รวมถึงกลุ่มองค์กรที่ต้องการใช้สนามและจัดกิจกรรมกีฬา (Figuresports [1]) นอกจากสนามกีฬาแล้ว บริษัท SportHub ยังพัฒนาเว็บไซต์ที่ช่วยอำนวยความสะดวกให้ผู้ใช้สามารถจองสนามออนไลน์ได้อย่างรวดเร็ว ลดความซับซ้อนของการติดต่อแบบเดิม อีกทั้งธุรกิจยังมีขอบเขตครอบคลุม การลงทะเบียนเข้าร่วมการแข่งขันกีฬา ต่าง ๆ และการเช่าอุปกรณ์กีฬาได้ในระบบเดียว (Figuresports [1]) SportHub มุ่งหวังเป็นแพลตฟอร์มกลางที่ช่วยเชื่อมโยงกลุ่มคนรักกีฬา และสร้างประสบการณ์ที่สุดสะดวกสบายและตอบโจทย์การใช้บริการด้านกีฬายุคใหม่

แผนผังหน้าต่างๆ ของเว็บแอปพลิเคชัน

แบบนำทางของเว็บไซต์ (Navigation Bar)



(ภาพที่ 1 แสดงแบบนำทางของเว็บไซต์)

1. Home Page

The screenshot shows the homepage of the Sport Hub website. At the top, there is a navigation bar with links for Facility, Tournament, Booking Check, Contact, a search bar, and buttons for Login and Sign Up. A red arrow labeled "Navigation" points to the top left of the page area. Below the navigation bar is a green header bar with the text "Welcome To Sport Hub !". Underneath this, there are two images: one of a soccer field and one of a tennis court. To the right of these images is a weather widget showing "Today's Weather" in Chiang Mai, Thailand, with a temperature of 32.7 °C. Below the weather is a text block about the facility, followed by social media icons. A red arrow labeled "Footer" points to the bottom left of the page area. At the very bottom, there is a map titled "We are here !" showing the location of the facility in Chiang Mai, Thailand.

(ภาพที่ 2 แสดงรายการและอีเมลหน้า Home)

2. “Login” Page

The image shows two side-by-side login pages. The left page is for "User" and the right page is for "Admin". Both pages have a "Log in using your account on:" section with "User" selected. A red arrow points from the "User" selection on the left to the "Admin" selection on the right, indicating they are the same. Both pages also have a "Sport Hub" link at the bottom.

(ภาพที่ 3 แสดงให้เห็นการเลือก log-in สำหรับ User และ Admin โดยเข็มไปยังหน้า log-in สำหรับ Admin)

3. “Search” page

The screenshot shows the 'Advanced Search' page of a sports facility booking system. At the top, there's a navigation bar with links for Facility, Tournament, Booking Check, Contact, a search bar, and user options (Login, Sign Up). Below the navigation is a form titled 'Advanced Search' with fields for Service (Facility), Name, Sport Type (Football), and Date (set to 06/06/2022). A red arrow points from the 'Search' button in this form to the search bar at the top of the page.

Facility Options:

- Football Field 1:** Type - Football, Description - Outdoor grass football field, Open/Close - 06:00:00 - 22:00:00, Status - Available. Buttons: Booking Now! (green), Booked (grey).
- Football Field 2:** Type - Football, Description - Outdoor grass football field, Open/Close - 06:00:00 - 22:00:00, Status - In Use. Buttons: Booking Now! (green), Booked (grey).
- Football Field 3:** Type - Football, Description - Outdoor grass football field, Open/Close - 06:00:00 - 22:00:00, Status - Closed. Buttons: Booking Now! (green), Booked (grey).

(ภาพที่ 4 แสดงรายละเอียดต่างๆในหน้า Search)

4. “Detail” page

The screenshot shows the detail page for 'Football Field 1'. At the top, there's a navigation bar with links for Facility, Tournament, Booking Check, Contact, a search bar, and user options (Login, Sign Up). Below the navigation is a large image of the football field and its details: Sport Type - Football, Description - Outdoor grass football field, Capacity - 30, Price/Person - 500 THB, Open/Close time - 06:00:00 - 22:00:00, and Status - Available. A green 'Booking Now!' button is present.

A red arrow points from the 'Booking Now!' button on the detail page back up to the 'Advanced Search' form on the previous page, indicating a direct link between the two.

Facility Options:

- Football Field 1:** Type - Football, Description - Outdoor grass football field, Open/Close - 06:00:00 - 22:00:00, Status - Available. Buttons: Booking Now! (green), Booked (grey).
- Football Field 2:** Type - Football, Description - Outdoor grass football field, Open/Close - 06:00:00 - 22:00:00, Status - In Use. Buttons: Booking Now! (green), Booked (grey).
- Football Field 3:** Type - Football, Description - Outdoor grass football field, Open/Close - 06:00:00 - 22:00:00, Status - Closed. Buttons: Booking Now! (green), Booked (grey).
- Badminton Court 1:** Type - Badminton, Description - Indoor court with lighting, Open/Close - 06:00:00 - 22:00:00, Status - Available. Buttons: Booking Now! (green), Booked (grey).
- Badminton Court 2:** Type - Badminton, Description - Indoor court with lighting, Open/Close - 06:00:00 - 22:00:00, Status - In Use. Buttons: Booking Now! (green), Booked (grey).
- Badminton Court 3:** Type - Badminton, Description - Indoor court with lighting, Open/Close - 06:00:00 - 22:00:00, Status - Closed. Buttons: Booking Now! (green), Booked (grey).
- Ping Pong Table 1:** Type - Ping Pong, Description - Indoor table with lighting, Open/Close - 06:00:00 - 22:00:00, Status - Available. Buttons: Booking Now! (green), Booked (grey).
- Ping Pong Table 2:** Type - Ping Pong, Description - Indoor table with lighting, Open/Close - 06:00:00 - 22:00:00, Status - In Use. Buttons: Booking Now! (green), Booked (grey).
- Ping Pong Table 3:** Type - Ping Pong, Description - Indoor table with lighting, Open/Close - 06:00:00 - 22:00:00, Status - Closed. Buttons: Booking Now! (green), Booked (grey).

(ภาพที่ 5 แสดงหน้ารายละเอียดของบริการ)

5. “Product/Service Management” page

The screenshot displays the Product/Service Management interface. On the left, a sidebar menu for 'Organization Management' lists categories: User, Facility, Equipment, Tournament, Booking History, and Rental History. The 'Facility' section is currently active, indicated by a red highlight. The main content area shows a table of facilities with columns: Facility ID, Facility Name, Sport Type, Price/Hour, Status, and Action. A red box highlights the 'Action' column, which contains buttons for 'Edit', 'Delete', and 'Details'. Below the table is a green 'Add New' button. To the right, a modal window titled 'Facility Edit (Football Field 1)' is open, showing fields for Facility Name (Football Field 1), Sport Type (Football), Price/Hour (100), Status (Available), and a large text area for Description. A red arrow points from the 'Edit' button in the table to this modal. Another red arrow points from the 'Edit' button in the modal back to the 'Edit' button in the table.

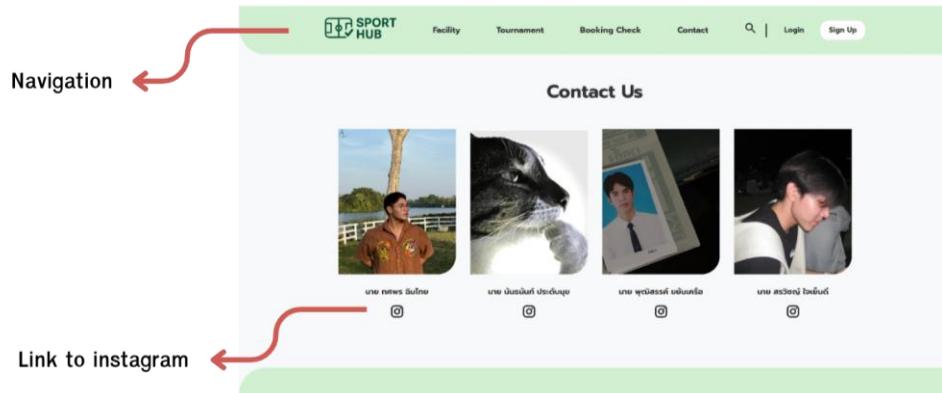
(ภาพที่ 6 แสดงหน้า Product/Service Management ที่สามารถแก้ไขข้อมูลเกี่ยวกับบริการ)

6. “User Account Management” page

The screenshot shows the User Account Management interface. The left sidebar menu for 'Organization Management' includes 'User' and 'Admin user' sections, with 'User' currently selected. The main area displays a table of users with columns: Username, E-mail, Role, and Action. A red box highlights the 'Action' column. To the right, a modal window titled 'Edit User (admin1)' is open, showing fields for Username (admin1), E-mail (admin1@sample.com), and Role (Admin). Red arrows point from the 'Edit' button in the table to the 'Edit' button in the modal, and from the 'Edit' button in the modal back to the 'Edit' button in the table. Below the table are two additional forms: 'Add User' and 'Add Admin User', both with various input fields and a green 'Save' button at the bottom.

(ภาพที่ 7 หน้า User Account Management ที่สามารถแก้ไขข้อมูลเกี่ยวกับ User และ Admin ได้)

7. “Team” page



(ภาพที่ 8 แสดงหน้า About us ซึ่งเป็นข้อมูลเกี่ยวกับสมาชิกและช่องทางการติดต่อ IG)

รายละเอียดหน้าเว็บต่างๆ

หน้าหลักของเว็บ (Home Page)

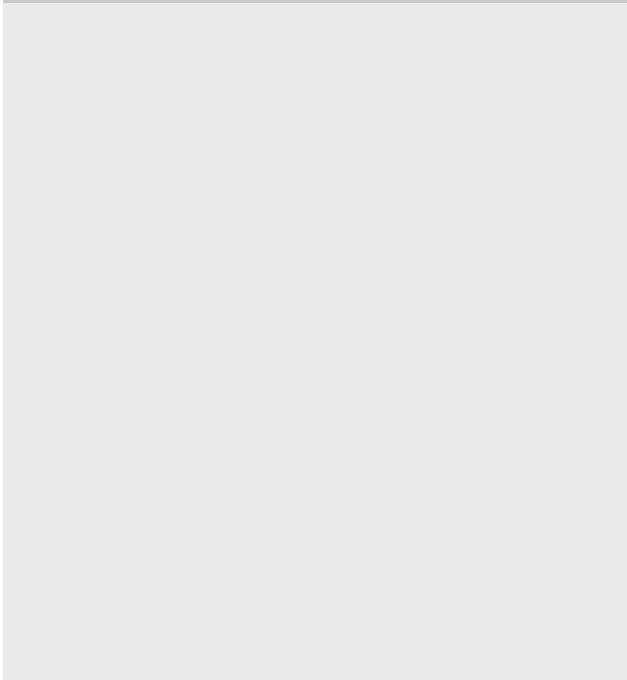
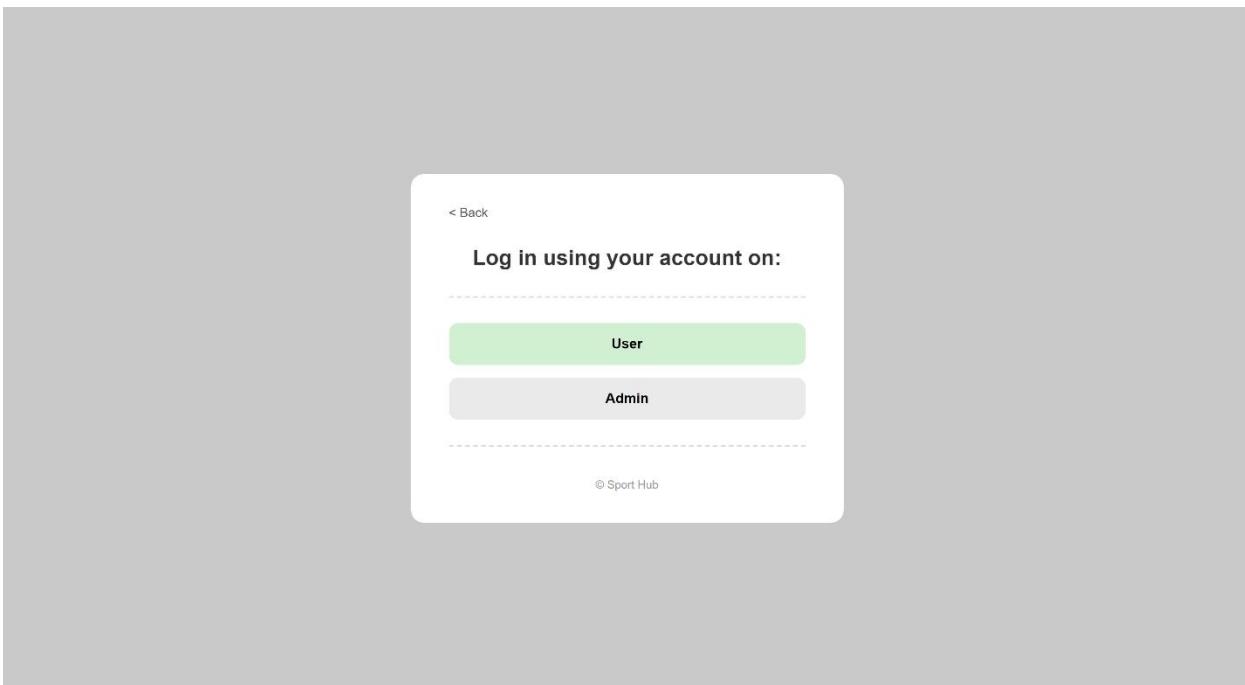
The screenshot shows the homepage of the Sport Hub website. At the top, there is a navigation bar with the logo "SPORT HUB", followed by links for "Facility", "Tournament", "Booking Check", "Contact", a search icon, "Login", and "Sign Up". Below the navigation bar, a large green banner with the text "Welcome To Sport Hub !" in white is displayed. Underneath the banner, there are two images: one of a soccer field at night and another of an indoor tennis court. To the right of these images is a weather forecast box showing "Today's Weather: เมฆเต็มฟ้า, 32.7 °C". Below the weather box is a paragraph of text about the facility, followed by a social media sharing section with icons for Instagram, Twitter, Facebook, and YouTube. At the bottom of the page is a map showing the location of the facility.

We are here !

A detailed Google Map of the area around the Sport Hub facility. The map shows several streets, buildings, and landmarks. Key locations labeled include "Figure Sports" (green dot), "AA Design" (blue dot), "Sincerity closet" (blue dot), "The Est Place" (pink dot), "บ้านวิเศษ" (yellow dot), "บ้านวิเศษ แม่ริม" (green dot), " MU Lake" (green dot), and "สุกี้" (green dot). The facility itself is marked with a blue dot and labeled "BOO1". The map also includes a "Google" watermark and a copyright notice "สงวนลิขสิทธิ์ ลิขสิทธิ์ ©2025 Google | สงวนลิขสิทธิ์".

ในหน้า Home เราจะมีส่วนบนของหน้าจอ มีแถบนำทาง (Navigation Bar) ซึ่งทำหน้าที่เชื่อมโยงผู้ใช้ไปยังหน้าสำคัญต่างๆ ของเว็บไซต์ โดยประกอบด้วยเมนูหลัก 4 เมนู คือ Facility (หน้ารวมสิ่งอำนวยความสะดวก), Tournament (หน้ารายการแข่งขัน), Booking Check (หน้าตรวจสอบการจอง), และ Contact (หน้าติดต่อ) และมี detail เกี่ยวกับบริการของเรา

หน้าเว็บสำหรับการเข้าสู่ระบบ (Log-in) ของผู้ดูแลระบบ



ในหน้า Log-in ของผู้ดูแลระบบ (Admin) จะต้องทำการเข้าสู่ระบบ โดยเลือกตัวเลือก 'Admin' ก่อน จากนั้นระบบจะให้ทำการกรอกชื่อผู้ใช้งาน (Username) และรหัสผ่าน (Password) หากข้อมูลที่กรอกมา ถูกต้องตรงตามที่บันทึกไว้ในฐานข้อมูล ผู้ดูแลระบบจะสามารถเข้าสู่หน้าจัดการเว็บไซต์ได้

หน้าสำหรับการค้นหาข้อมูลเว็บในส่วนสินค้าหรือเซอร์วิส

The screenshot shows the Sport Hub website's search interface. At the top, there is a navigation bar with links for Facility, Tournament, Booking Check, Contact, a search icon, Login, and Sign Up. Below the navigation bar, the title "Advanced Search" is displayed. The search form includes fields for Service (Facility selected), Name, Sport Type (Football selected), Date (mm/dd/yyyy format), and a "Search" button. Below the search form, three football fields are listed:

足球場	รายละเอียด
Football Field 1	Type : Football Description : Outdoor grass football field Open/Close : 06:00:00 - 22:00:00 Status : Available Booking Now!
Football Field 2	Type : Football Description : Outdoor grass football field Open/Close : 06:00:00 - 22:00:00 Status : In Use Booking Now!
Football Field 3	Type : Football Description : Outdoor grass football field Open/Close : 06:00:00 - 22:00:00 Status : Closed

ในหน้า Search (การค้นหา) ผู้ใช้สามารถใส่ ชื่อสนาม หรือ ชื่อรายการแข่งขัน เพื่อค้นหาสิ่งอำนวยความสะดวกที่ต้องการได้ หรือหากต้องการค้นหาที่ละเอียดกว่าเดิม จะมีตัวเลือกการค้นหาขั้นสูง (Advanced Search) โดยระบุ ประเภทบริการ (Service), ประเภทกีฬา (Sport Type), วันที่ (Date) ที่ต้องการ เมื่อระบุเงื่อนไขที่ต้องการแล้ว ผู้ใช้สามารถกดปุ่ม Search เพื่อแสดงผลลัพธ์ของ สนามหรือบริการ ที่ตรงตามเงื่อนไข

หน้าสำหรับการแสดงรายละเอียดสินค้าหรือเซอร์วิส

The screenshot shows a website for a sports facility. At the top, there is a navigation bar with the logo "SPORT HUB", followed by links for "Facility", "Tournament", "Booking Check", and "Contact". There are also "Login" and "Sign Up" buttons, along with a search icon.

The main content area displays a listing for "Football Field 1". It features a large image of a well-lit football pitch at night. To the right of the image, the title "Football Field 1" is displayed, along with several details:

- Sport Type :** Football
- Description :** Outdoor grass football field
- Capacity :** 30
- PricePerHour :** 500 THB
- Open-close time :** 06:00:00 - 22:00:00
- Status :** Available

A green "Booking Now!" button is located at the bottom of the listing.

ในหน้า Detail ของเว็บไซต์จะแสดงรายละเอียดของข้อมูลสินค้าและบริการของเรา โดยจะแสดงข้อมูล เช่น ชื่อ, ประเภทกีฬา, คำอธิบาย, ราคา, และสถานการณ์ใช้งานหรือกำหนดการซึ่งผู้ใช้สามารถกดย้อนกลับไปยังหน้า ก่อนหน้านี้ได้

หน้าสำหรับการจัดการข้อมูลของสินค้าหรือเซอร์วิส

Admin user

Facility Name	Sport Type	Status
Facility Name	Sport Type	Status

No. Facility ID Facility Name Type Price/Hour Status Actions

01	F0000001	Football Field 1	Football	500	Available	Edit Delete
02	F0000002	Football Field 2	Football	500	Reserved	Edit Delete
03	F0000003	Football Field 3	Football	500	Closed	Edit Delete
04	F0000004	Badminton Court 1	Badminton	300	Available	Edit Delete
05	F0000005	Badminton Court 2	Badminton	300	Reserved	Edit Delete
06	F0000006	Badminton Court 3	Badminton	300	Closed	Edit Delete
07	F0000007	Ping Pong Table 1	Ping Pong	120	Available	Edit Delete
08	F0000008	Ping Pong Table 2	Ping Pong	120	Reserved	Edit Delete
09	F0000009	Ping Pong Table 3	Ping Pong	120	Closed	Edit Delete

+ Add New

Admin user

< Back

Facility Edit (Football Field 1)

Facility Name	Football Field 1
Sport type	Football
Description	Outdoor grass football field
Capacity (People)	5
PricePerHour	500
OpenTime	06:00 AM
CloseTime	10:00 PM
Status	Available

Confirm

Add New Facility

Facility Name
e.g. Football Field 4

Sport type
Select sport type

Description
Details about the facility

Capacity (People)
e.g. 100

PricePerHour (THB)
e.g. 500

OpenTime

CloseTime

Status
Available

Confirm Add

No.	Facility ID	Facility Name	Type	Price/Hour	Status	Actions
01	F0000001	Football Field 1	Football	500	● Available	
02	F0000002	Football Field 2	Football	500	● In Use	
03	F0000003	Football Field 3	Football	500	● Closed	
04	F0000004	Badminton Court 1	Badminton	300	● Available	
05	F0000005	Badminton Court 2	Badminton	300	● In Use	
06	F0000006	Badminton Court 3	Badminton	300	● Closed	
07	F0000007	Ping Pong Table 1	Ping Pong	120	● Available	
08	F0000008	Ping Pong Table 2	Ping Pong	120	● In Use	
09	F0000009	Ping Pong Table 3	Ping Pong	120	● Closed	

+ Add New

ในหน้า Manage Facility นั้นเราจะแสดงรายละเอียดต่างๆ ของ Facility เช่น Facility_ID, Facility_Name, Type, Price/hour, Status ผู้ดูแลระบบสามารถใช้งาน ช่องค้นหา (Search) เพื่อจำกัดผลลัพธ์ตาม Facility_Name, Type และ Status ได้อย่างรวดเร็ว นอกจากนี้ยังมาร์คฟังก์ชันการจัดการข้อมูลหลัก (CRUD) โดยผู้ดูแลระบบสามารถ เพิ่ม (Create): Facility ใหม่ ผ่านแบบฟอร์มได้ แก้ไข (Update): แก้ไขรายละเอียดของรายการที่มีอยู่ และ ลบ (Delete) Facility ที่ไม่ใช้งานออกจากระบบได้ทันที

หน้าสำหรับการจัดการข้อมูลผู้ดูแลระบบ

User Management

No.	User ID	Username	E-mail	BirthDate	Role	Actions
01	U0000001	admin1	admin1@gmail.com	2005-05-10	Staff	<button>Edit</button> <button>Delete</button>
02	U0000002	somjai	somjai@gmail.com	1995-11-20	User	<button>Edit</button> <button>Delete</button>
03	U0000003	somying	somying@gmail.com	1998-03-05	User	<button>Edit</button> <button>Delete</button>
04	A0000001	superadmin	admin1@gmail.com	1985-11-20	Admin	<button>Edit</button> <button>Delete</button>
05	A0000002	equipmgr	chai.mgr@gmail.com	2000-09-17	Admin	<button>Edit</button> <button>Delete</button>
06	A0000003	facmgr	nicha.mgr@gmail.com	2001-01-02	Admin	<button>Edit</button> <button>Delete</button>

[+ Add User](#) [+ Add Admin user](#)

Edit User (admin1)

First Name: Somchai

Last Name: Deemak

Phone Number: 0890000001

Email: admin1@gmail.com

Gender: Other

Address: Bangkok

Date of Birth: 05/10/2005

Role: Staff

[Confirm Edit](#)

Add Admin User

Username:

Password:

First Name:

Last Name:

Phone Number:

Email:

Gender:

Address:

Date of Birth: mm / dd / yyyy

Status:

Confirm Add Admin

localhost:3000 says
คุณต้องการลบผู้ใช้ที่ ID U0000001 ?
(คุณจะไม่สามารถย้อนกลับได้)

No.	User ID	Username	E-mail	BirthDate	Role	Actions
01	U0000001	Staff1	admin1@gmail.com	2005-05-10	Staff	<button>Edit</button> <button>X Delete</button>
02	U0000002	somjai	somjai@gmail.com	1995-11-20	User	<button>Edit</button> <button>X Delete</button>
03	U0000003	somying	somying@gmail.com	1998-03-05	User	<button>Edit</button> <button>X Delete</button>
04	A0000001	superadmin	admin1@gmail.com	1985-11-20	Admin	<button>Edit</button> <button>X Delete</button>
05	A0000002	equipmgr	chai.mgr@gmail.com	2000-09-17	Admin	<button>Edit</button> <button>X Delete</button>
06	A0000003	facmgr	nicha.mgr@gmail.com	2001-01-02	Admin	<button>Edit</button> <button>X Delete</button>

+ Add User **+ Add Admin user**

ในหน้า Manage user ระบบจะแสดง User_ID, User_Name, Email, BirthDate, Role ผู้ดูแลระบบ
สามารถใช้งานช่องค้นหา (Search) เพื่อจำกัดผลลัพธ์ตาม User_Name, Email และ Role ได้อย่างรวดเร็ว
นอกจากนี้ ผู้ดูแลระบบยังมีสิทธิในการจัดการบัญชีผู้ใช้งานหลัก (User) Staff และบัญชีผู้ดูแลระบบ (Admin
User) โดยมีฟังก์ชันการจัดการข้อมูลหลัก (CRUD) เพิ่ม (Create): สามารถเพิ่มบัญชี User หรือ Admin ใหม่ได้
แก้ไข (Update): สามารถแก้ไขข้อมูลส่วนตัวได้ และ ลบ (Delete): สามารถลบบัญชีออกจากระบบได้

หน้า About Us แสดงรายละเอียดสมาชิกในกลุ่ม

The screenshot shows a website header with the logo "SPORT HUB" and navigation links for Facility, Tournament, Booking Check, Contact, a search bar, Login, and Sign Up. Below the header is a section titled "Contact Us" featuring four profile cards:

- นาย กศพร อินโภัย (นาย กศพร อินโภัย)
- นาย บันเต้นบันท ประดับบุญ (นาย บันเต้นบันท ประดับบุญ)
- นาย พุดเนสรรค์ บอยบันเกร็ง (นาย พุดเนสรรค์ บอยบันเกร็ง)
- นาย สรวิษณ์ ใจเย็นดี (นาย สรวิษณ์ ใจเย็นดี)

Each card includes a small profile picture and a social media icon below it.

ในหน้า About Us จะแสดงรูป, ชื่อ และ ช่องทางการติดต่อ (IG) ของสมาชิกในกลุ่มทุกคน

รายละเอียดของเว็บเซอร์วิส และโค้ด

1. เว็บเซอร์วิส หน้า Landing.html

CODE	<pre>app.get('/', (req, res) => { // path.join จะหาไฟล์ Landing.html ในโฟลเดอร์ปัจจุบัน res.sendFile(path.join(__dirname, 'Landing.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path '/' (root) โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Landing.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>document.addEventListener('DOMContentLoaded', function() { // ○○○ API Key ○○○ const MY_API_KEY = 'b29b2b69d7c924ab3c3226e1d718c81e'; // พิกัดของ "พื้นที่สนามกีฬา" const lat = 13.792047274471141; const lon = 100.31197178580933; const weatherWidget = document.getElementById('weather-widget'); // สร้าง URL สำหรับเรียก API (ขอหน่วย Metric (C) และภาษาไทย) const apiUrl = `https://api.openweathermap.org/data/2.5/weather?lat=\${lat}&lon=\${lon}&appid=\${MY_API_KEY}&units=metric&lang=th`; // --- 2. ใช้ API ด้วย Fetch --- fetch(apiUrl) .then(response => { if (!response.ok) { throw new Error('Network response was not ok'); } return response.json(); // แปลงข้อมูลเป็น JSON }) .then(data => { // --- 3. นำข้อมูลมาแสดงผล --- console.log(data); // คุณสามารถดูข้อมูลทั้งหมดที่ได้ใน Console const temp = data.main.temp; // อุณหภูมิ(C)</pre>
------	--

```

        const weatherDescription =
    data.weather[0].description; // คำอธิบาย
        const weatherIcon = data.weather[0].icon; // ไอคอน
        const iconUrl =
`https://openweathermap.org/img/wn/${weatherIcon}@2x.png`;

        // สร้าง HTML ใหม่เพื่อแสดงผล
        weatherWidget.innerHTML =
            <strong>Today's Weather:</strong>
            
            <span>${weatherDescription}, ${temp.toFixed(1)}
        °C</span>
            `;
        })
        .catch(error => {
            console.error('Error fetching weather:', error);
            weatherWidget.innerHTML = '<p>ไม่สามารถโหลดข้อมูลอากาศ
        ได้</p>';
        });
    });

```

คำอธิบาย	เว็บเซอร์วิสสาธารณะ โดยใช้เพื่อบอกสภาพอากาศ ณ จุดนั้นๆ
-----------------	--

2. เว็บเซอร์วิส หน้า Facility.html

CODE	<pre>app.get('/facility', (req, res) => { res.sendFile(path.join(__dirname, 'Facility.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path '/facility' โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Facility.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>router.get('/facilities', (req, res) => { const { name, type, status } = req.query; // เริ่มต้น SQL let sql = 'SELECT * FROM Facility WHERE 1=1'; const params = []; // เพิ่มไข้การค้นหา(ถ้ามี) if (name) { sql += ' AND Name LIKE ?'; params.push(`%\${name}%`); } if (type) { sql += ' AND Type LIKE ?'; params.push(`%\${type}%`); } if (status) { sql += ' AND Status LIKE ?'; params.push(`%\${status}%`); } sql += ' ORDER BY Facility_ID ASC'; db.query(sql, params, (err, results) => { if (err) { console.error('Error fetching facilities:', err); return res.status(500).json({ success: false, message: 'Database error' }); } res.json(results); }); });</pre>
------	--

คำอธิบาย	กำหนด Route แบบ HTTP GET ที่ path /facilities เพื่อค้นหาและดึงข้อมูลสนามกีฬา โดยรองรับการกรองข้อมูลตามชื่อ (Name), ประเภท (Type) และสถานะ (Status) และส่งคืนผลลัพธ์เป็น JSON
-----------------	--

3. เว็บเซอร์วิส หน้า Tournament.html

CODE	<pre>app.get('/tournament', (req, res) => { res.sendFile(path.join(__dirname, 'Tournament.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path '/tournament' โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Tournament.html.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 1. GET /tournaments (Search) --- router.get('/tournaments', (req, res) => { const { name, type, status } = req.query; // ⚡ ขั้นตอน 1: Tournaments -> Tournament let sql = 'SELECT Tournament_ID, Name, SportType, Status FROM Tournament WHERE 1=1'; const searchParams = []; if (name) { sql += " AND Name LIKE ?"; searchParams.push(`%\${name}%`); } if (type) { sql += " AND SportType LIKE ?"; searchParams.push(`%\${type}%`); } if (status) { sql += " AND Status LIKE ?"; searchParams.push(`%\${status}%`); } sql += " ORDER BY Tournament_ID ASC"; db.query(sql, searchParams, (err, results) => { if (err) return res.status(500).json({ success: false, message: 'Database error' }); res.json(results); }); });</pre>
-------------	---

	<pre> }); }); </pre>
คำอธิบาย	<p>กำหนด Route แบบ HTTP GET ที่ path /tournaments เพื่อค้นหารายการแข่งขัน (Tournament) โดยสามารถกรองข้อมูลตามชื่อรายการ, ประเภทกีฬา และสถานะการแข่งขัน ได้ ระบบจะส่งคืนข้อมูลที่ตรงตามเงื่อนไขกลับไป</p>

\

4. เว็บเซอร์วิส หน้า Contact.html

CODE	<pre>app.get('/contact', (req, res) => { res.sendFile(path.join(__dirname, 'Contact.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /contact โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Contact.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

5. เว็บเซอร์วิส หน้า Search.html

CODE	<pre>app.get('/search', (req, res) => { res.sendFile(path.join(__dirname, 'Search.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /search โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Search.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>router.get('/search/tournament', async (req, res) => { try { const { name, type, date } = req.query; // ตั้งค่า Table 'Tournament' let sql = 'SELECT * FROM Tournament WHERE 1=1'; const params = []; if (name) { sql += ' AND Name LIKE ?'; // params.push(`%\${name}%`); } if (type) { sql += ' AND SportType = ?'; // params.push(type); } if (date) { // ตั้งค่า 'date' อยู่ระหว่าง 'StartDate' ถึง 'EndDate' sql += ' AND ? BETWEEN StartDate AND EndDate'; params.push(date); } } catch (err) { console.error(err); res.status(500).send('Internal Server Error'); } res.json({sql, params}); });</pre>
------	---

	<pre> } // --- [↓] ແຈ້ນມາໃຫ້ Callback ຕຽງນີ້ [↓] --- dbPool.execute(sql, params, (error, rows) => { if (error) { // ຊ້າ Query ນີ້ບໍ່ສູ່ຫາ console.error('Error executing query:', error); res.status(500).json({ error: 'Failed to search tournaments' }); return; // ຈະການທຳງານ } // ຊ້າສໍາເລັດ ສົ່ງຂໍ້ມູນ 'rows' ກັບໄປ res.json(rows); }); // --- [↑] ສິ້ນສຸດຄ່າວນ Callback [↑] --- } catch (error) { // Error ອີ່ຈະບັນແພະຄວາມຄືດພາດກ່ອນທີ່ຈະຮັນ dbPool.execute console.error('Error searching tournaments:', error); res.status(500).json({ error: 'Failed to search tournaments' }); } }); </pre>
ຄໍາອອິບາຍ	ກຳໜາດ Route ສໍາຫຼັບກາຣົນຫາຂັ້ນສູງ (Advanced Search) ຂໍອມູນກາຣແຂ່ງຂັ້ນ ໂດຍຮອງຮັບ ເຈື້ອນໄຂຊື່ອ, ປະເກທກີ່ພາ ແລະ ຕຽບສອບວ່າວັນທີທີ່ຮະບູ (Date) ອູ້ໃນໆໃໝ່ເວລາຂອງກາຣແຂ່ງຂັ້ນ හີ່ວ່າມີ

CODE	<pre> router.get('/search/facility', async (req, res) => { try { const { name, type } = req.query; // ຊັນຫາທາງ Table 'Facility' let sql = 'SELECT * FROM Facility WHERE 1=1'; } }); </pre>
------	--

	<pre> const params = []; if (name) { sql += ' AND Name LIKE ?'; // params.push(`%\${name}%`); } if (type) { sql += ' AND Type = ?'; // params.push(type); } // --- ⬇️ ပေါ်ယောນမာໃ့Callback တရှုနှံ ⬇️ --- dbPool.execute(sql, params, (error, rows) => { if (error) { // ဦးQuery မိုးသွား console.error('Error executing query:', error); res.status(500).json({ error: 'Failed to search facilities' }); return; // ပုံမှန်ခံခြင်း } // ဦးဆားသွေးစံချိန် 'rows' ကလုပ်ပါပဲ res.json(rows); }); // --- ⬆️ ဆီမံခွင့်လာမာ Callback ⬆️ --- } catch (error) { // Error ဖူးချိန်အပဲအမာမိမိပေါ်လာတဲ့အတွက် console.error('Error searching facilities:', error); res.status(500).json({ error: 'Failed to search facilities' }); } }); </pre>
คำอธิบาย	กำหนด Route สำหรับการค้นหาขั้นสูง (Advanced Search) ข้อมูลสนามกีฬา โดยค้นหาจากชื่อและประเภทของสนามตามที่ผู้ใช้ระบุ และส่งคืนผลลัพธ์กลับเป็น JSON

6. เว็บเซอร์วิส หน้า Choose_login.html

CODE	<pre> app.get('/choose_login', (req, res) => { res.sendFile(path.join(__dirname, 'Choose_login.html')); }); </pre>
------	---

คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /choose_login โดยมีคำขอเข้ามา เชิร์ฟเวอร์จะส่งไฟล์ Choose_login.html กลับไปยัง client เพื่อแสดงหน้าเว็บ
-----------------	--

7. เว็บเซอร์วิส หน้า Admin_login.html

CODE	<pre>app.get('/admin_login', (req, res) => { res.sendFile(path.join(__dirname, 'Admin_login.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /admin_login โดยมีคำขอเข้ามา เชิร์ฟเวอร์จะส่งไฟล์ Admin_login.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 1. Admin Login --- router.post('/admin/login', (req, res) => { const { username, password } = req.body; const sql = 'SELECT * FROM Admin WHERE username = ? AND password = ?'; db.query(sql, [username, password], (err, results) => { if (err) return res.status(500).json({ success: false, message: 'Server error' }); if (results.length > 0) { const admin = results[0]; res.json({ success: true, message: 'Login successful', adminId: admin.Admin_ID, role: admin.Role, name: admin.FName + ' ' + admin.LName }); } else { res.json({ success: false, message: 'Username หรือ Password ไม่ถูกต้อง' }); } }); });</pre>
คำอธิบาย	กำหนด Route แบบ HTTP POST ที่ path /admin/login เพื่อตรวจสอบการเข้าสู่ระบบของผู้ดูแลระบบ โดยนำ Username และ Password ไปตรวจสอบในฐานข้อมูล หากถูกต้องจะส่งคืนสถานะสำเร็จพร้อมข้อมูล Admin

8. เว็บเซอร์วิส หน้า manageuser.html

CODE	<pre>app.get('/manageuser', (req, res) => { res.sendFile(path.join(__dirname, 'manageuser.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path '/manageuser' โดยเมื่อเมื่อคำขอเข้ามา เซิร์ฟเวอร์ จะส่งไฟล์ manageuser.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 2. Get Users & Admins (Search) --- router.get('/users', (req, res) => { const { username, email, role } = req.query; let conditionSql = ""; const searchParams = []; if (username) { conditionSql += " AND username LIKE ?"; searchParams.push(`%\${username}%`); } if (email) { conditionSql += " AND Email LIKE ?"; searchParams.push(`%\${email}%`); } if (role) { conditionSql += " AND Role LIKE ?"; searchParams.push(`%\${role}%`); } let sql = ` SELECT User_ID, username, Email, Role, DateOfBirth, CreatedAt FROM users WHERE 1=1 \${conditionSql} UNION SELECT Admin_ID as User_ID, username, Email, Role, DateOfBirth, CreatedAt FROM Admin WHERE 1=1 \${conditionSql} ORDER BY CreatedAt DESC `; const finalParams = [...searchParams, ...searchParams]; // นี่คือ Params สำหรับ UNION db.query(sql, finalParams, (err, results) => { if (err) return res.status(500).json({ success: false, message: 'Database error' }); res.json(results); }); });</pre>
------	--

คำอธิบาย	กำหนด Route เพื่อดึงข้อมูลบัญชีผู้ใช้ทั้งหมด โดยใช้คำสั่ง UNION เชื่อมข้อมูลจากตาราง Users และ Admin เข้าด้วยกัน รองรับการค้นหาตามชื่อผู้ใช้, อีเมล หรือบทบาท (Role)
-----------------	--

CODE	<pre>// --- 3. Get Single User/Admin by ID ---(For Edit) router.get('/users/:id', (req, res) => { const id = req.params.id; let sql = ''; if (id.startsWith('U')) { sql = 'SELECT * FROM users WHERE User_ID = ?'; } else if (id.startsWith('A')) { sql = 'SELECT Admin_ID as User_ID, username, password, FName, LName, Email, Tel, Role, Status, Address, Gender, DateOfBirth, CreatedAt FROM Admin WHERE Admin_ID = ?'; } else { return res.status(400).json({ message: 'Invalid ID format' }); } db.query(sql, [id], (err, results) => { if (err) return res.status(500).json({ error: err.message }); if (results.length === 0) return res.status(404).json({ message: 'User/Admin not found' }); res.json(results[0]); }); });</pre>
คำอธิบาย	กำหนด Route เพื่อดึงข้อมูลรายละเอียดของผู้ใช้รายบุคคลตาม ID โดยระบบจะตรวจสอบรหัสชี้นำต้น (Prefix) เพื่อแยกແຍະและดึงข้อมูลจากตาราง Users หรือ Admin ได้อย่างถูกต้อง

CODE	<pre>// --- 7. Delete User/Admin (DELETE) --- router.delete('/users/:id', (req, res) => { const id = req.params.id; let sql = ''; if (id.startsWith('U')) { sql = 'DELETE FROM users WHERE User_ID = ?'; } else if (id.startsWith('A')) { sql = 'DELETE FROM Admin WHERE Admin_ID = ?'; } else { return res.status(400).json({ success: false, message: 'Invalid ID format' }); } db.query(sql, [id], (err, result) => { if (err) return res.status(500).json({ success: false, message: 'Cannot delete (Foreign Key constraint)' }); if (result.affectedRows === 0) return res.status(404).json({ success: false, message: 'Data not found' }); res.json({ success: true, message: 'Deleted successfully' }); }); });</pre>
คำอธิบาย	กำหนด Route แบบ HTTP DELETE เพื่อลบบัญชีผู้ใช้งานออกจากระบบ โดยตรวจสอบว่าเป็น User หรือ Admin จากรหัส ID ก่อนดำเนินการลบข้อมูลในตารางที่เกี่ยวข้อง

9. เว็บเซอร์วิส หน้า AddUser.html

CODE	<pre>app.get('/adduser', (req, res) => { res.sendFile(path.join(__dirname, 'AddUser.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path '/adduser' โดยเมื่อป้อนคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ AddUser.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 4. Create User (POST) --- router.post('/users', (req, res) => { const { username, password, FName, LName, Tel, Email, Gender, Address, DateOfBirth, Role, Status } = req.body; db.query('SELECT MAX(User_ID) as maxId FROM users', (err, results) => { if (err) return res.status(500).json({ success: false, message: err.message }); let newUserId = 'U0000001'; if (results[0].maxId) { const currentMax = parseInt(results[0].maxId.substring(1)); const nextId = currentMax + 1; newUserId = 'U' + nextId.toString().padStart(7, '0'); } const sql = `INSERT INTO users (User_ID, username, password, FName, LName, Email, Tel, Gender, DateOfBirth, Address, Role, Status) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)`; const values = [newUserId, username, password, FName, LName, Email, Tel, Gender, DateOfBirth, Address, Role, Status]; db.query(sql, values, (insertErr) => { if (insertErr) { if (insertErr.code === 'ER_DUP_ENTRY') return res.status(400).json({ success: false, message: 'Username นี้ถูกใช้งานแล้ว' }); return res.status(500).json({ success: false, message: 'เพิ่มข้อมูลไม่สำเร็จ: ' + insertErr.message }); } }); }); });</pre>
------	--

	<pre> res.json({ success: true, message: 'เพิ่มผู้ใช้งานเร็ว' }); }); }); }); </pre>
คำอธิบาย	กำหนด Route แบบ HTTP POST สำหรับสร้างบัญชีผู้ใช้งานใหม่ (User) โดยระบบจะทำการคำนวนและสร้าง User ID ลำดับถัดไปให้อัตโนมัติ (Auto-generate ID) และบันทึกข้อมูลลงฐานข้อมูล

10. เว็บเซอร์วิส หน้า AddAdmin html

CODE	<pre>app.get('/addadmin', (req, res) => { res.sendFile(path.join(__dirname, 'AddAdmin.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /addadmin โดยเมื่อมีคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ AddAdmin.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 5. Create Admin (POST) --- router.post('/admins', (req, res) => { const { username, password, FName, LName, Tel, Email, Gender, Address, DateOfBirth, Status } = req.body; db.query('SELECT MAX(Admin_ID) as maxId FROM Admin', (err, results) => { if (err) return res.status(500).json({ success: false, message: err.message }); let newAdminId = 'A0000001'; if (results[0].maxId) { const currentMax = parseInt(results[0].maxId.substring(1)); const nextId = currentMax + 1; newAdminId = 'A' + nextId.toString().padStart(7, '0'); } const sql = `INSERT INTO Admin (Admin_ID, username, password, FName, LName, Email, Tel, Gender, DateOfBirth, Address, Role, Status) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, 'Admin', ?)`; const values = [newAdminId, username, password, FName, LName, Email, Tel, Gender, DateOfBirth, Address, Status]; db.query(sql, values, (insertErr) => { if (insertErr) { if (insertErr.code === 'ER_DUP_ENTRY') return res.status(400).json({ success: false, message: 'Username นี้ถูกใช้งานแล้ว' }); return res.status(500).json({ success: false, message: 'เพิ่ม Admin ไม่สำเร็จ: ' + insertErr.message }); } res.json({ success: true, message: 'เพิ่ม Admin สำเร็จ' }); }); }); });</pre>
------	---

	<pre> }); }); </pre>
คำอธิบาย	กำหนด Route แบบ HTTP POST สำหรับสร้างบัญชีผู้ดูแลระบบใหม่ (Admin) โดยระบบจะทำการสร้าง Admin ID ลำดับถัดไปให้อัตโนมัติ และบันทึกข้อมูลลงในตาราง Admin

11. เว็บเซอร์วิส หน้า Edituser.html

CODE	<pre>app.get('/edituser', (req, res) => { res.sendFile(path.join(__dirname, 'Edituser.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /edituser โดยเมื่อป้อนคำขอเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Edituser.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 6. Update User/Admin (PUT) --- router.put('/users/:id', (req, res) => { const id = req.params.id; const { FName, LName, Tel, Email, Gender, Address, DateOfBirth, Role } = req.body; // CASE 1: Users (ID starts with U) - กรณี Logic Promote Admin if (id.startsWith('U')) { db.query('SELECT * FROM users WHERE User_ID = ?', [id], (err, results) => { if (err results.length === 0) return res.status(404).json({ message: 'User not found' }); const currentUser = results[0]; const updateSql = `UPDATE users SET FName=?, LName=?, Tel=?, Email=?, Gender=?, Address=?, DateOfBirth=?, Role=? WHERE User_ID=?`; const updateValues = [FName, LName, Tel, Email, Gender, Address, DateOfBirth, Role, id]; db.query(updateSql, updateValues, (err) => { if (err) return res.status(500).json({ success: false, message: 'Update failed' }); if (Role === 'Admin') { db.query('SELECT * FROM Admin WHERE username = ?', [currentUser.username], (err, adminCheck) => { if (adminCheck.length === 0) { db.query('SELECT MAX(Admin_ID) as maxId FROM Admin', (err, idResult) => { let newAdminId = 'A0000001'; if (idResult[0].maxId) { const currentMax = parseInt(idResult[0].maxId.substring(1)); } } } } } }) } });</pre>
------	--

```

        const nextId = currentMax + 1;
        newAdminId = 'A' +
nextId.toString().padStart(7, '0');
    }
    const insertAdminSql = `INSERT INTO Admin (Admin_ID, username, password, FName, LName, Email, Tel, Role, Status) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, 'Active')`;
    db.query(insertAdminSql,
[newAdminId, currentUser.username, currentUser.password, FName, LName, Email, Tel, 'Admin']);
}
} else {
    db.query('DELETE FROM Admin WHERE username = ?',
[currentUser.username]);
}
res.json({ success: true, message: 'User updated successfully' });
});
});
}
// CASE 2: Admin (ID starts with A)
else if (id.startsWith('A')) {
    const sql = `UPDATE Admin SET FName=?, LName=?, Tel=?, Email=?, Role=?, Address=?, Gender=?, DateOfBirth=? WHERE Admin_ID=?`;
    const values = [FName, LName, Tel, Email, Role, Address, Gender, DateOfBirth, id];

    db.query(sql, values, (err) => {
        if (err) return res.status(500).json({ success: false, message: 'Update Admin failed' });
        res.json({ success: true, message: 'Admin updated successfully' });
    });
}
});

```

คำอธิบาย	กำหนด Route แบบ HTTP PUT เพื่อแก้ไขข้อมูลผู้ใช้งาน โดยรองรับทั้ง User และ Admin และมีฟังก์ชันพิเศษในการเลื่อนสถานะ (Promote) จาก User เป็น Admin โดยจะเพิ่มข้อมูลลงตาราง Admin ให้อตโนมัติ
----------	--

12. เว็บเซอร์วิส หน้า Mainfacility.html

CODE	<pre>app.get('/mainfacility', (req, res) => { res.sendFile(path.join(__dirname, 'Mainfacility.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /mainfacility โดยเมื่อคิ้มขอเข้ามา เซิร์ฟเวอร์ จะส่งไฟล์ Mainfacility.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 1. GET: ค้นหาและดึงข้อมูล Facility (Search) --- router.get('/facilities', (req, res) => { const { name, type, status } = req.query; // รีบดู SQL let sql = 'SELECT * FROM Facility WHERE 1=1'; const params = []; // เงื่อนไขการค้นหา (ถ้ามี) if (name) { sql += ' AND Name LIKE ?'; params.push(`%\${name}%`); } if (type) { sql += ' AND Type LIKE ?'; params.push(`%\${type}%`); } if (status) { sql += ' AND Status LIKE ?'; params.push(`%\${status}%`); } sql += ' ORDER BY Facility_ID ASC'; db.query(sql, params, (err, results) => { if (err) { console.error('Error fetching facilities:', err); return res.status(500).json({ success: false, message: 'Database error' }); } res.json(results); }); });</pre>
------	---

คำอธิบาย	กำหนด Route สำหรับหน้าจัดการสนามกีฬา เพื่อค้นหาและแสดงรายชื่อสนามทั้งหมด โดยสามารถกรองข้อมูลตามชื่อ, ประเภท และสถานะ เพื่อนำมาแสดงผลในตารางจัดการ
-----------------	---

CODE	<pre>// --- 3. DELETE: ลบ Facility --- router.delete('/facilities/:id', (req, res) => { const id = req.params.id; db.query('DELETE FROM Facility WHERE Facility_ID = ?', [id], (err, result) => { if (err) return res.status(500).json({ success: false, message: 'Cannot delete (Foreign Key constraint)' }); if (result.affectedRows === 0) return res.status(404).json({ success: false, message: 'Data not found' }); res.json({ success: true, message: 'Deleted successfully' }); }); });</pre>
คำอธิบาย	กำหนด Route แบบ HTTP DELETE ที่ path /facilities/:id เพื่อทำการลบข้อมูลสนามกีฬาที่ระบุออกจากฐานข้อมูล

CODE	<pre>// --- 4. GET: ดึงข้อมูล Facility 1 แล้ว(By ID) สำหรับหน้า Edit --- router.get('/facilities/:id', (req, res) => { const id = req.params.id; db.query('SELECT * FROM Facility WHERE Facility_ID = ?', [id], (err, results) => { if (err) return res.status(500).json({ success: false, message: 'Database error' }); if (results.length === 0) return res.status(404).json({ success: false, message: 'Not found' }); res.json(results[0]); }); });</pre>
คำอธิบาย	กำหนด Route เพื่อดึงข้อมูลรายละเอียดของสนามกีฬา 1 แห่งตาม ID ที่ระบุ เพื่อนำข้อมูลเดิมมาแสดงผลในแบบฟอร์มก่อนการแก้ไข

13. เว็บเซอร์วิส หน้า Addnewfacility

CODE	<pre>app.get('/addnewfacility', (req, res) => { res.sendFile(path.join(__dirname, 'Addnewfacility.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /addnewfacility โดยเมื่อมีคำขอเข้ามา เชิร์ฟเวอร์จะส่งไฟล์ Addnewfacility.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

CODE	<pre>// --- 2. POST: เพิ่ม Facility ใหม่(Auto Generate ID) --- router.post('/facilities', (req, res) => { const { Name, Type, Description, Capacity, OpenTime, CloseTime, PricePerHour, Status } = req.body; db.query('SELECT MAX(Facility_ID) as maxId FROM Facility', (err, results) => { if (err) return res.status(500).json({ success: false, message: err.message }); let newId = 'F0000001'; if (results[0].maxId) { const currentMax = parseInt(results[0].maxId.substring(1)); const nextId = currentMax + 1; newId = 'F' + nextId.toString().padStart(7, '0'); } const sql = `INSERT INTO Facility (Facility_ID, Name, Type, Description, Capacity, OpenTime, CloseTime, PricePerHour, Status) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)`; const values = [newId, Name, Type, Description, Capacity, OpenTime, CloseTime, PricePerHour, Status]; db.query(sql, values, (err) => { if (err) return res.status(500).json({ success: false, message: 'Insert failed: ' + err.message }); res.json({ success: true, message: 'Facility added successfully', id: newId }); }); }); });</pre>
------	---

คำอธิบาย	กำหนด Route แบบ HTTP POST สำหรับเพิ่มสนามกีฬาใหม่ โดยระบบจะทำการสร้างรหัสสนาม (Facility ID) ลำดับถัดไปให้อัตโนมัติ และบันทึกรายละเอียดต่างๆ ลงในฐานข้อมูล
-----------------	---

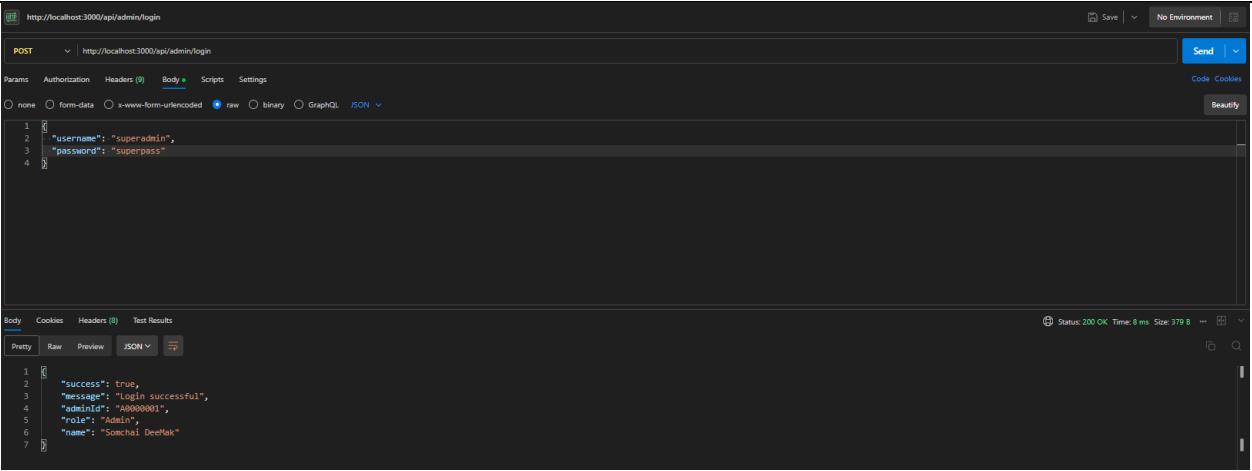
14. เว็บเซอร์วิส หน้า Editfacility.html

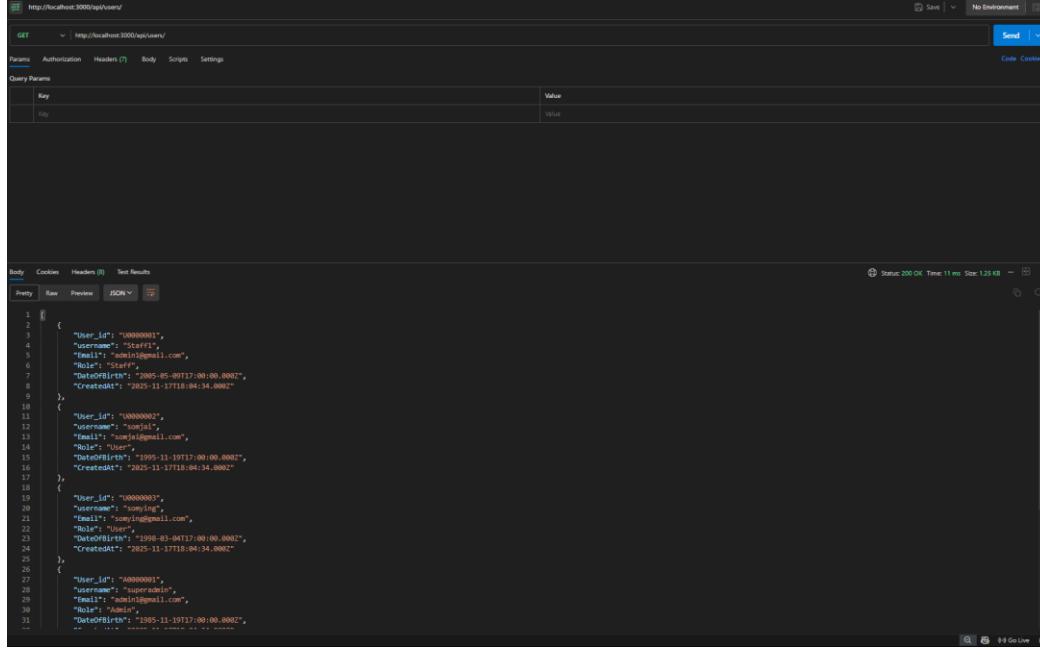
CODE	<pre>app.get('/editfacility', (req, res) => { res.sendFile(path.join(__dirname, 'Editfacility.html')); });</pre>
คำอธิบาย	กำหนด Route สำหรับ HTTP GET ที่ path /editfacility โดยเมื่อเมื่อเข้ามา เซิร์ฟเวอร์จะส่งไฟล์ Editfacility.html กลับไปยัง client เพื่อแสดงหน้าเว็บ

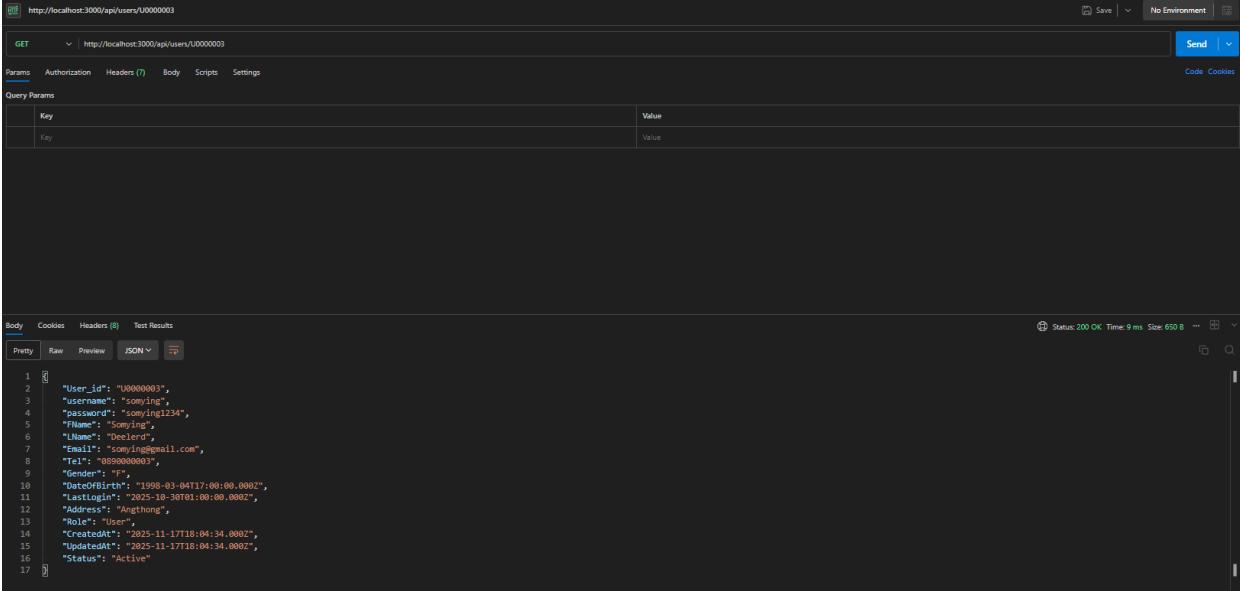
CODE	<pre>// --- 5. PUT: แก้ไขข้อมูล Facility --- router.put('/facilities/:id', (req, res) => { const id = req.params.id; const { Name, Type, Description, Capacity, OpenTime, CloseTime, PricePerHour, Status } = req.body; const sql = `UPDATE Facility SET Name=?, Type=?, Description=?, Capacity=?, OpenTime=?, CloseTime=?, PricePerHour=?, Status=? WHERE Facility_ID=?`; const values = [Name, Type, Description, Capacity, OpenTime, CloseTime, PricePerHour, Status, id]; db.query(sql, values, (err) => { if (err) return res.status(500).json({ success: false, message: 'Update failed: ' + err.message }); res.json({ success: true, message: 'Facility updated successfully' }); }); });</pre>
คำอธิบาย	กำหนด Route แบบ HTTP PUT ที่ path /facilities/:id เพื่อบันทึกการแก้ไขข้อมูลรายละเอียดของสนามกีฬาตาม ID ที่ระบุลงในฐานข้อมูล

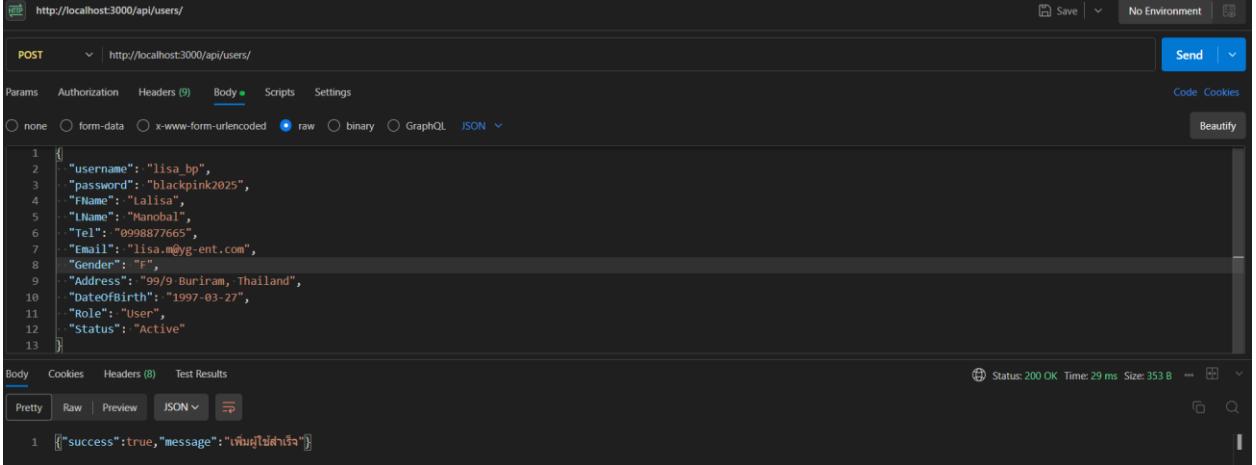
ผลการทดสอบของเว็บเซอร์วิสทั้งหมด โดยใช้โปรแกรม Postman

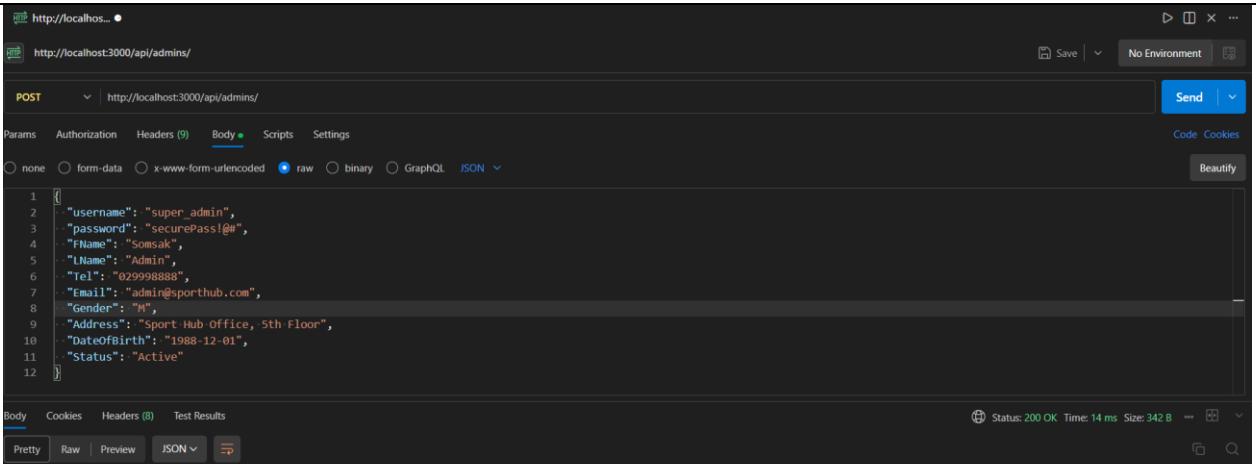
UserRoutes

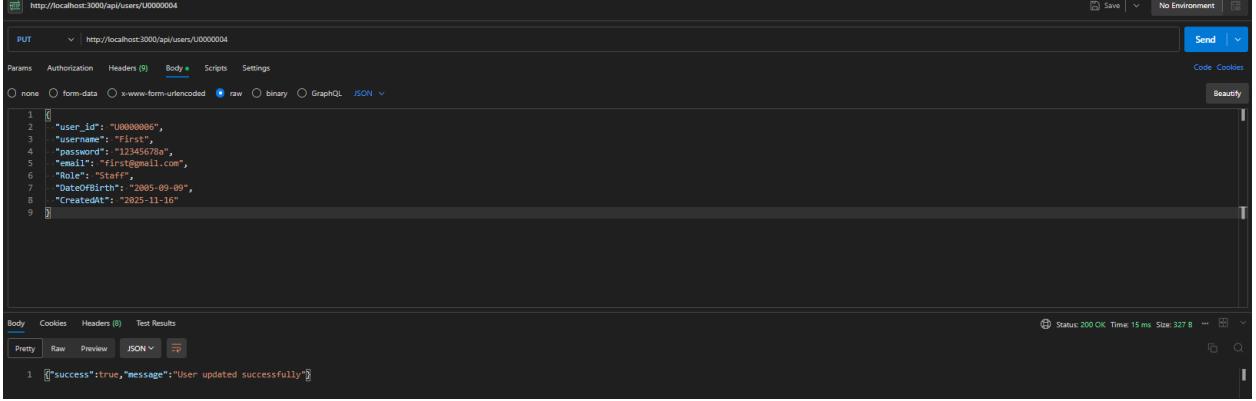
URL	http://localhost:3000/api/admin/login
Method	POST
Body	 <pre>POST http://localhost:3000/api/admin/login</pre> <pre>Params Authorization Headers (9) Body Scripts Settings</pre> <pre>none form-data x-www-form-urlencoded raw binary GraphQL JSON</pre> <pre>1 { 2 "username": "superadmin", 3 "password": "superpass" 4 }</pre> <pre>Body Cookies Header (8) Test Results</pre> <pre>Pretty Raw Preview JSON</pre> <pre>1 { 2 "success": true, 3 "message": "Login successful", 4 "adminId": "A0000001", 5 "role": "Admin", 6 "name": "Somchai DeeMak" 7 }</pre> <p>Status: 200 OK Time: 8 ms Size: 379 B</p>
Description	การ Login ของ Admin

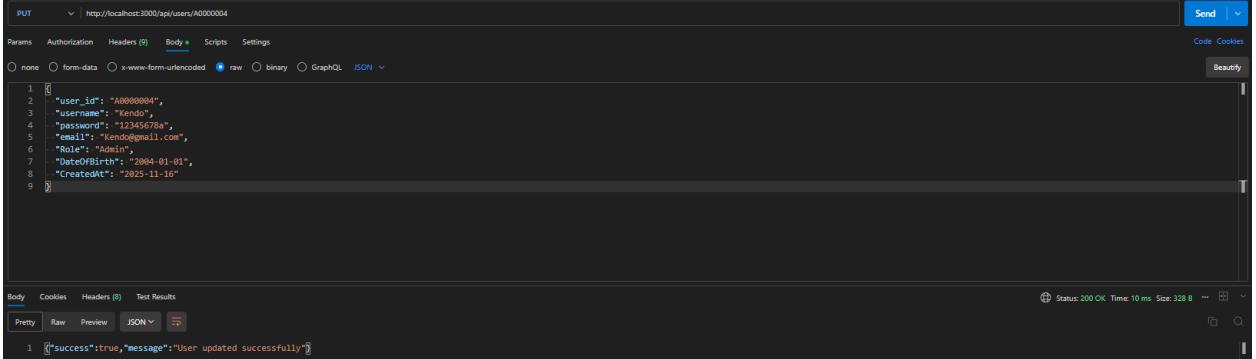
URL	http://localhost:3000/api/users/
Method	GET
Body	 <pre> 1 [2 { 3 "User_Id": "U0000001", 4 "username": "staf#1", 5 "Email": "admin@gmail.com", 6 "Role": "User", 7 "DateOfBirth": "2005-05-00T17:00:00.000Z", 8 "CreatedAt": "2023-11-17T18:04:14.000Z" 9 }, 10 { 11 "User_Id": "U0000002", 12 "username": "soyding", 13 "Email": "soyding@gmail.com", 14 "Role": "User", 15 "DateOfBirth": "1995-11-19T17:00:00.000Z", 16 "CreatedAt": "2023-11-17T18:04:14.000Z" 17 }, 18 { 19 "User_Id": "U0000003", 20 "username": "soyding", 21 "Email": "soyding@gmail.com", 22 "Role": "User", 23 "DateOfBirth": "1998-03-04T17:00:00.000Z", 24 "CreatedAt": "2023-11-17T18:04:14.000Z" 25 }, 26 { 27 "User_Id": "A0000001", 28 "username": "Admin", 29 "Email": "admin@gmail.com", 30 "Role": "Admin", 31 "DateOfBirth": "1995-11-19T17:00:00.000Z" 32 } 33] </pre>
Description	ดึงข้อมูล User และ Admin ตามที่ค้นหา ถ้าไม่ส่อไรเลยจะดึงข้อมูลทั้งหมด

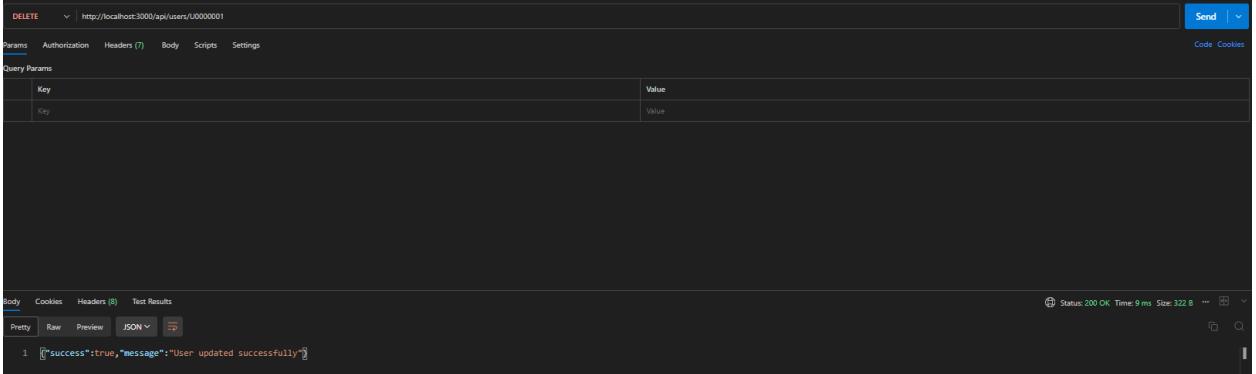
URL	http://localhost:3000/api/users/U0000003
Method	GET
Body	 <pre> 1 { 2 "User_id": "U0000003", 3 "username": "somying", 4 "password": "somying1234", 5 "FName": "Somying", 6 "LName": "Angthong", 7 "Email": "somying@gmail.com", 8 "Tel": "0899000003", 9 "Gender": "F", 10 "DateOfBirth": "1998-03-04T17:00:00.000Z", 11 "LastLogin": "2025-10-30T01:00:00.000Z", 12 "Address": "Angthong", 13 "Role": "User", 14 "CreatedAt": "2025-11-17T18:04:34.000Z", 15 "UpdatedAt": "2025-11-17T18:04:34.000Z", 16 "Status": "Active" 17 } </pre>
Description	ดึงข้อมูล User หรือ Admin ตาม User ID

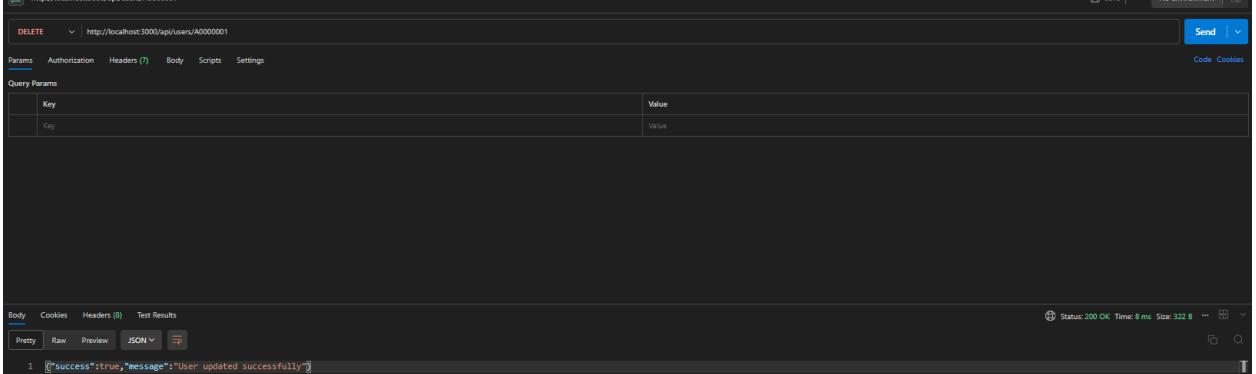
URL	http://localhost:3000/api/users/
Method	POST
Body	 <pre> 1 { 2 "username": "lisa_bp", 3 "password": "blackpink2025", 4 "fName": "Lalisa", 5 "lName": "Manoban", 6 "tel": "0998877665", 7 "email": "lisa.m@y-g-ent.com", 8 "gender": "F", 9 "address": "99/9 Buriram, Thailand", 10 "dateOfBirth": "1997-03-27", 11 "role": "User", 12 "status": "Active" 13 } </pre> <p>Status: 200 OK Time: 29 ms Size: 353 B</p> <p>["success":true,"message":"เพิ่มผู้ใช้สำเร็จ"]</p>
Description	เพิ่มบัญชี User ใหม่

URL	http://localhost:3000/api/admins/
Method	POST
Body	 <pre> 1 { 2 "username": "super_admin", 3 "password": "securePass123", 4 "fName": "Somsak", 5 "lName": "Admin", 6 "tel": "029998888", 7 "email": "admin@porthub.com", 8 "gender": "M", 9 "address": "Sport Hub Office, 5th Floor", 10 "dateOfBirth": "1988-12-01", 11 "status": "Active" 12 } </pre> <p>Status: 200 OK Time: 14 ms Size: 342 B</p> <p>["success":true,"message":"เพิ่ม Admin สำเร็จ"]</p>
Description	เพิ่มบัญชี Admin

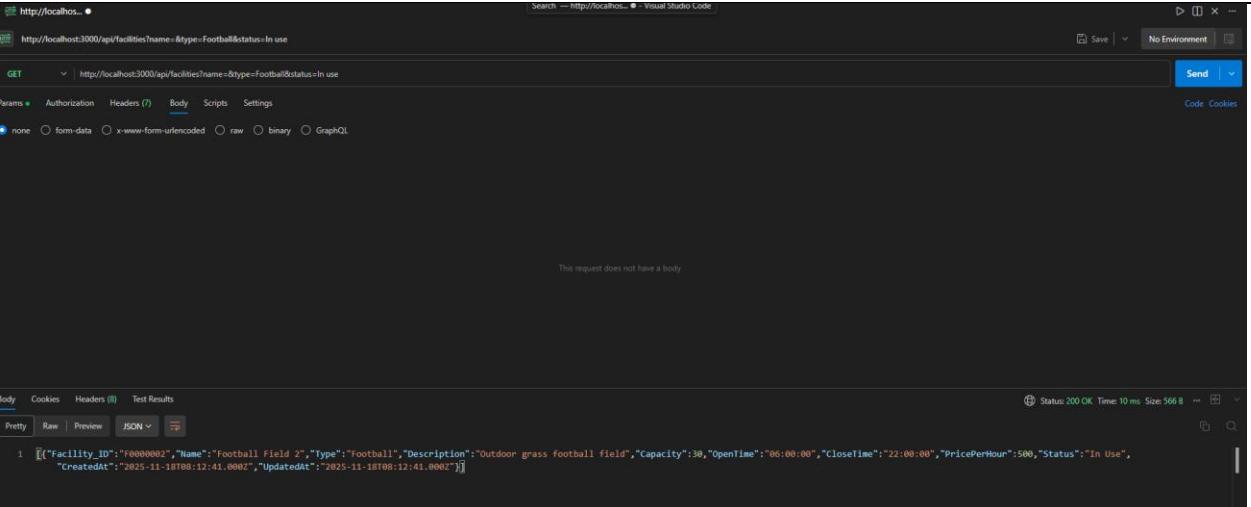
URL	http://localhost:3000/api/users/U0000004
Method	PUT
Body	 <pre> 1 [2 "user_id": "U0000004", 3 "username": "First", 4 "password": "12345678a", 5 "email": "First@gmail.com", 6 "role": "Staff", 7 "DateOfBirth": "2005-09-09", 8 "CreatedAt": "2025-11-16" 9] </pre> <p>Status: 200 OK Time: 15 ms Size: 327 B</p> <pre> 1 {"success":true,"message":"User updated successfully"} </pre>
Description	แก้ไขข้อมูล User

URL	http://localhost:3000/api/users/A0000004
Method	PUT
Body	 <pre> 1 [2 "user_id": "A0000004", 3 "username": "Kendo", 4 "password": "12345678a", 5 "email": "Kendo@gmail.com", 6 "role": "Admin", 7 "DateOfBirth": "2004-01-01", 8 "CreatedAt": "2025-11-16" 9] </pre> <p>Status: 200 OK Time: 10 ms Size: 328 B</p> <pre> 1 {"success":true,"message":"User updated successfully"} </pre>
Description	แก้ไขข้อมูล Admin

URL	http://localhost:3000/api/users/U0000001
Method	Delete
Body	 <pre>1 {"success":true,"message":"User updated successfully"}</pre>
Description	ลบบัญชี User

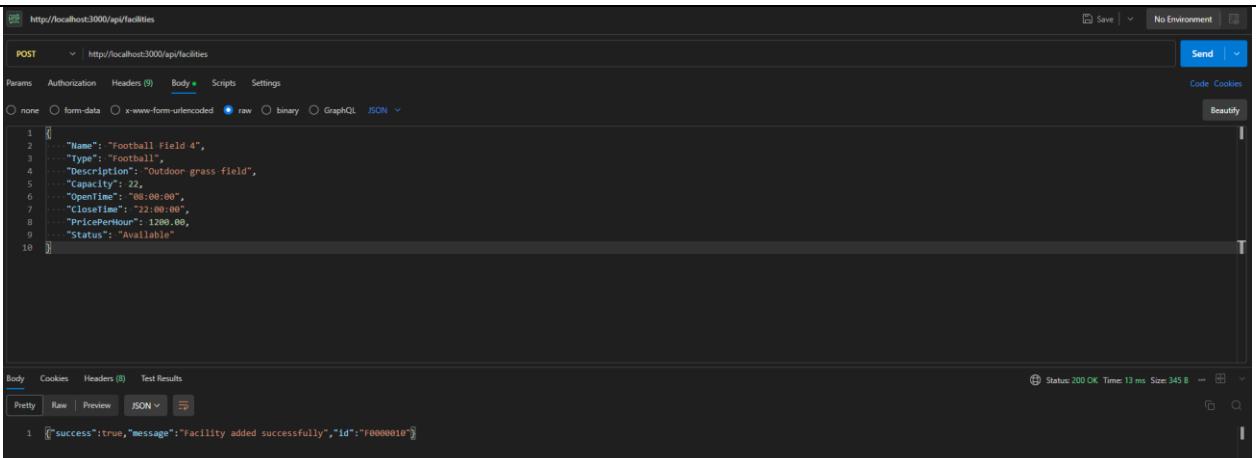
URL	http://localhost:3000/api/users/A0000001
Method	Delete
Body	 <pre>1 {"success":true,"message":"User updated successfully"}</pre>
Description	ลบบัญชี Admin

FacilityRoutes

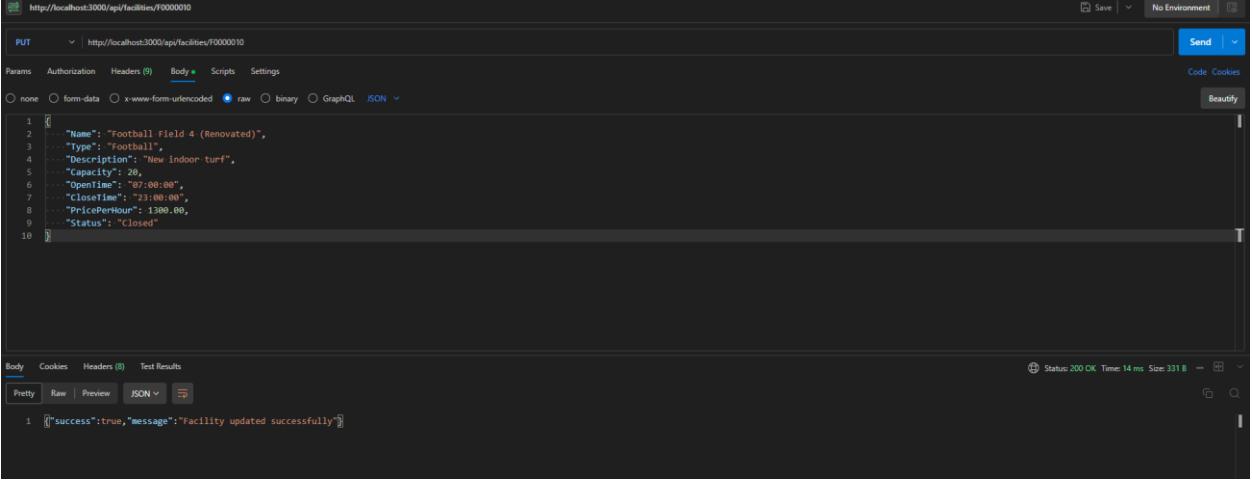
URL	http://localhost:3000/api/facilities
Method	GET
Body	 <pre> 1 [{"Facility_ID": "F0000002", "Name": "Football Field 3", "Type": "Football", "Description": "Outdoor grass football field", "Capacity": 30, "OpenTime": "06:00:00", "CloseTime": "22:00:00", "PricePerHour": 500, "Status": "In Use", "CreatedAt": "2025-11-18T08:12:41.000Z", "UpdatedAt": "2025-11-18T08:12:41.000Z"}] </pre>
Description	ดึงข้อมูล Facilities ได้ตามที่ค้นหา ถ้าไม่ใส่ค่าค้นหาจะดึงข้อมูลทั้งหมด

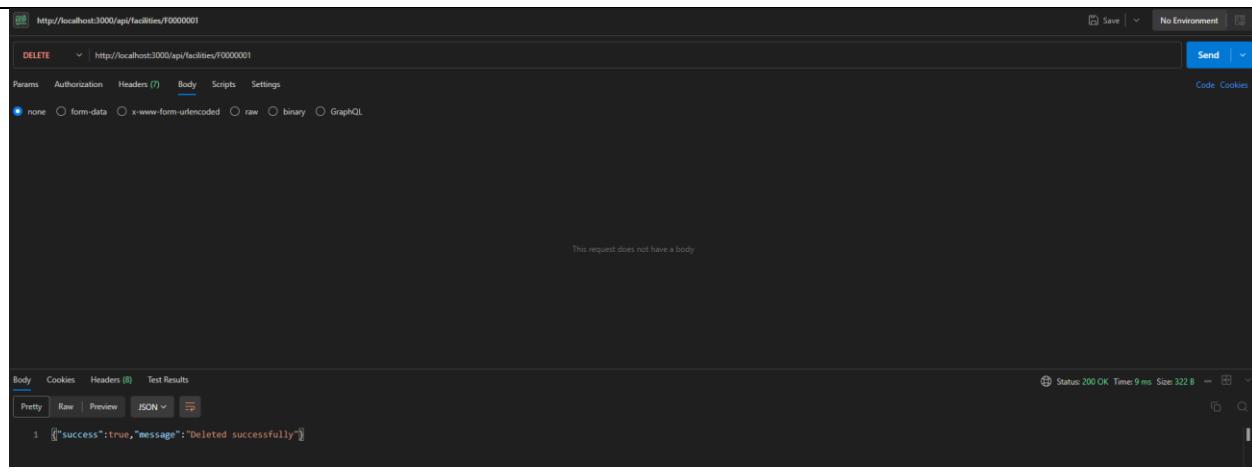
URL	http://localhost:3000/api/facilities/F0000001
Method	GET
Body	 <pre> 1 [{"User_ID": "U0000001", 2 ... "Facility_ID": "F0000002", 3 ... "StartDateTime": "2025-11-21T15:00:00", 4 ... "EndDateTime": "2025-11-21T17:00:00", 5 ... "Status": "Confirmed" 6 } 7] </pre> <p>Status: 200 OK Time: 9 ms Size: 567 B</p>

Description ดึงข้อมูล Facility อันเดียว

URL	http://localhost:3000/api/users/
Method	POST
Body	 <pre> 1 [{"Name": "Football Field 4", 2 ... "Type": "football", 3 ... "Description": "Outdoor grass field", 4 ... "Capacity": 30, 5 ... "OpenTime": "06:00:00", 6 ... "CloseTime": "22:00:00", 7 ... "PricePerHour": 1200.00, 8 ... "Status": "Available" 9 } 10] </pre> <p>Status: 200 OK Time: 13 ms Size: 345 B</p>

Description เพิ่ม Record Facility ใหม่

URL	http://localhost:3000/api/users/U0000004
Method	PUT
Body	 <pre> 1 { 2 "Name": "Football Field 4 (Renovated)", 3 "Type": "Football", 4 "Description": "New indoor turf", 5 "Capacity": 20, 6 "OpenTime": "07:00:00", 7 "CloseTime": "23:00:00", 8 "PricePerHour": 100.00, 9 "Status": "Closed" 10 } </pre> <p>Status: 200 OK Time: 14 ms Size: 331 B</p>
Description	ແກ້ໄຂຂໍ້ມູນ Facility

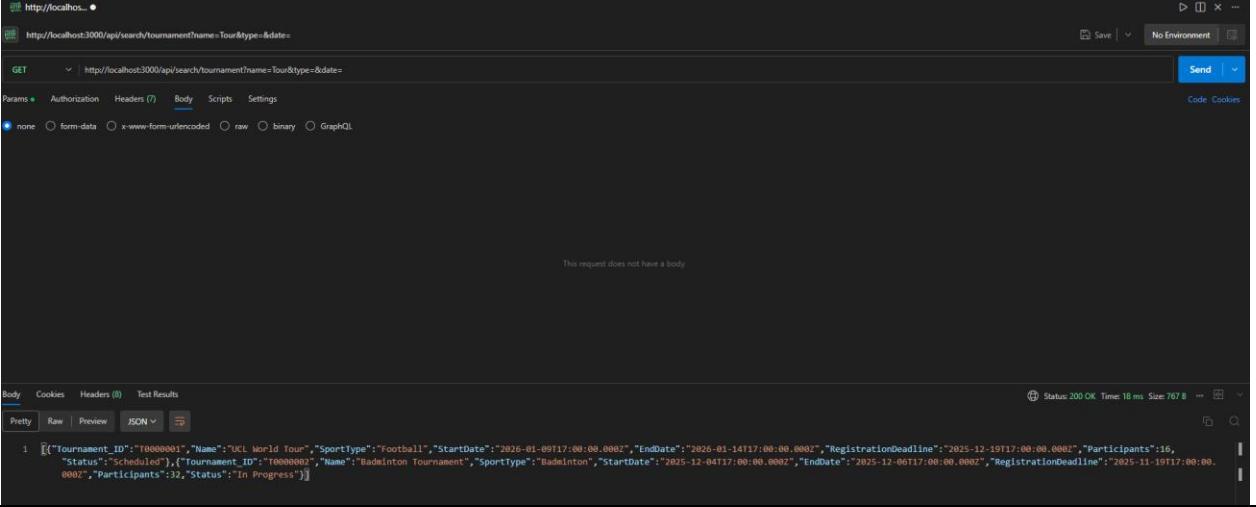
URL	http://localhost:3000/api/facilities/F0000001
Method	Delete
Body	 <p>This request does not have a body</p> <p>Status: 200 OK Time: 9 ms Size: 322 B</p>
Description	ລົບບັນຫຼື User

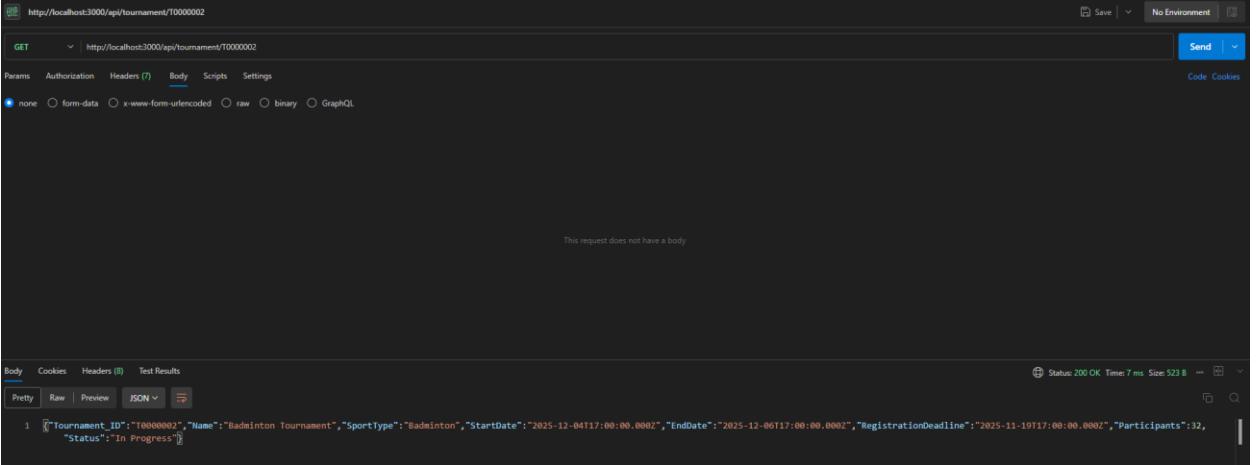
searchingfacilityRoutes

URL	http://localhost:3000/api/search/facility?name=1&type=Football
Method	GET
Body	
Description	ดึงข้อมูล Facility ตามที่ค้นหา ถ้าไม่ได้จะถึงข้อมูลทั้งหมด

URL	http://localhost:3000/api/facility/F0000003
Method	GET
Body	
Description	ดึงข้อมูล Facility อันเดียว

searchingtournamentRoute

URL	http://localhost:3000/api/search/tournament?name=Tour&type=&date=
Method	GET
Body	 <pre> [{"Tournament_ID": "T0000001", "Name": "UCL World Tour", "SportType": "Football", "StartDate": "2026-01-09T17:00:00.000Z", "EndDate": "2026-01-14T17:00:00.000Z", "RegistrationDeadline": "2025-12-19T17:00:00.000Z", "Participants": 16, "Status": "Scheduled"}, {"Tournament_ID": "T0000002", "Name": "Badminton Tournament", "SportType": "Badminton", "StartDate": "2025-12-04T17:00:00.000Z", "EndDate": "2025-12-06T17:00:00.000Z", "RegistrationDeadline": "2025-11-19T17:00:00.000Z", "Participants": 32, "Status": "In Progress"}] </pre>
Description	ดึงข้อมูล Facility ตามที่ค้นหา ถ้าไม่ได้จะดึงข้อมูลทั้งหมด

URL	http://localhost:3000/api/tournament/T0000002
Method	GET
Body	 <pre> [{"Tournament_ID": "T0000002", "Name": "Badminton Tournament", "SportType": "Badminton", "StartDate": "2025-12-04T17:00:00.000Z", "EndDate": "2025-12-06T17:00:00.000Z", "RegistrationDeadline": "2025-11-19T17:00:00.000Z", "Participants": 32, "Status": "In Progress"}] </pre>
Description	ดึงข้อมูล Facility อันเดียว

ข้อความเปิดเผยการใช้เทคโนโลยีปัญญาประดิษฐ์

กลุ่มของข้าพเจ้าขอรับรองว่าได้มีการใช้เทคโนโลยี Generative AI ในการจัดทำกิจกรรมหรือชิ้นงานนี้โดยนำเครื่องมือ Generative AI (ChatGPT, Gemini) มาใช้เพื่อวัตถุประสงค์ดังต่อไปนี้ :

- ระดมความคิด
- ร่างชิ้นงานเบื้องต้น
- ตรวจสอบและแก้ไขภาษา ไวยากรณ์ หรือการแปล
- สรุปและถอดความเนื้อหา
- แก้ไขและตรวจสอบโค้ด
- เขียนโค้ด
- ใช้ AI เป็นส่วนหนึ่งในผลงาน
- อื่น ๆ (โปรดระบุ): _____

แหล่งอ้างอิง

เว็บไซต์

- [1] Figuresports. [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568. <https://figuresports.net/>
- [2] รูปสนามบล. [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568. <https://wipeselectric.com/>
- [3] รูปสนามแบดต. [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568. <https://pbsport.co.th/portfolio-page/all-star-badminton/>
- [4] รูปสนามปิงปอง. [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568.
<https://www.pingponghouse.com/17109735>
- [5] รูปโปสเตอร์การแข่งขันฟุตบอล [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568.
https://www.freepik.com/premium-vector/football-tournament_264169221.htm
- [6] รูปโปสเตอร์การแข่งขันแบดมินตัน [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568.
<https://www.shutterstock.com/image-vector/badminton-tournament-poster-sports-event-2527115697>
- [7] รูปโปสเตอร์การแข่งขันปิงปอง [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568.
<https://th.postermywall.com/index.php/art/template/cc3d0e75b83b27c6098e7026487be7a6/ping-pong-tournament-design-template>
- [8] เว็บจองสนามกีฬา มหาวิทยาลัยวไลลักษณ์. [ระบบออนไลน์]. สืบค้นเมื่อ 10 กันยายน 2568.
<https://booksport.wu.ac.th/index.php?r=booking>

ภาคผนวก

ต้นแบบการออกแบบเว็บไซต์ (Figma)

<https://www.figma.com/design/oUP0Mi6QQ81yNJewd0ZzlH/Project-web?m=auto&t=RAC1wmcFOfmQcOmf-1>