**PROG2002 – Web Development II**

**Assignment 2: Use Case (A Dynamic Website)**

**Student ID:24516785** **Last Name:Liu** **First Name:Haonan**

**Title of the project:Charity activity management platform**

**Introduction/Motivation**

As society's attention to public welfare keeps increasing, charitable activities are playing an increasingly important role in communities. However, many charitable organizations are confronted with problems such as poor dissemination of event information and complex registration processes for participants. This project aims to develop a dynamic website to provide an efficient and user-friendly event management platform for charitable organizations and the public.

**Problem Statement**

1. Information on charity activities is scattered, making it difficult for the public to obtain complete event information

2. The traditional registration method is inefficient and lacks a unified registration platform

3. It is difficult for event organizers to effectively manage event information and participant data

4. The lack of real-time update and search functions results in a poor user experience

**Solution**

This project adopts a client-server architecture to develop a complete dynamic website:

Front-end: Use HTML, CSS, JavaScript and DOM operations

The back end: RESTful apis are provided using Node.js and the Express framework

Database: Use MySQL to store active data

Communication: Data interaction between the front-end and back-end is achieved through AJAX

**Web UX**

1. Intuitive navigation structure ensures that users can easily find the information they need

2. Responsive design, compatible with screens of different devices

3. Clear visual hierarchy, highlighting important information

4. Smooth interactive feedback enhances user participation

**Data Schema**

1.Home Page: Organization Information + Activity List

2. Search page: Multi-condition filtering form + result display

3. Detail Page: Complete event information + registration function

Data model:

categories (Category Table)

├── youdaoplaceholder5 (primary key)

└ -- name (Category Name)

events (Event Schedule

├── event\_id (primary key)

├── title (Activity Title)

├── description (Brief description)

├── event\_date (Event Date)

├── location (Event Location

├── youdaoplaceholder5 (foreign key)

── ticket\_price (Ticket price)

├── goal\_amount (Target amount)

├── current\_amount (Current amount)

├── is\_active

├── full\_description (Complete description)

└── created\_at (Creation time)

**API design**

1. GET /api/events

Purpose: Get all active upcoming events

HTTP method: GET

Request subject: None

Responding subject

{

"event\_id": 1,

title: "Beach Sunset Fun Run 2025"

"description": "Family-friendly running activity along the beach"

"event\_date": "2025-10-10",

"location": "Bondi Beach"

"category\_name": "Fun Run"

"ticket\_price": 25.00

}

Method selection basis: Obtain resource data, in line with the semantics of RESTful GET

2. GET /api/events/search

Purpose: Search for activities based on conditions

HTTP method: GET

Request parameters:

date (optional): Event date

location (Optional): Location keyword

category (Optional): Category ID

Response body: The same as /api/events

Method selection basis: Query operation, use GET to pass query parameters

3. GET /api/events/:id

Purpose: To obtain details of individual events

HTTP method: GET

Request parameter: id (path parameter)

Response subject: Complete information of a single activity

Method selection basis: Obtain specific resources and use path parameters to identify the resources

4. GET /api/categories

Purpose: To obtain all activity categories

HTTP method: GET

Request subject: None

Responding subject

{" youdaoplaceholder5 ": 1, "name": "Fun run "}

{" youdaoplaceholder5 ": 2, "name": "Charity dinner "}

Method selection basis: Obtain the resource list, in line with RESTful design