



**** ข้อกำหนด ****

นักศึกษาต้องออกแบบ **end-point (uri)** ของแต่ละ **resource** เอง โดยทุก ๆ end-point ต้องมี a-รหัสนักศึกษาสามตัวท้ายของนศ. เป็นส่วนประกอบด้วยเสมอ เช่น <http://localhost:8080/a-001/api/subjects> เป็นต้น

ต้องเขียนโปรแกรมตาม **Layer System Architectures (controller-service-data access)**

รูปแบบของชื่อฟิลด์ **JSON** ใน request/response ต้องเป็น **Camel Case**

1. จาก ER Diagram ให้นักศึกษา สร้าง Express Project เพื่อให้บริการ api ตามข้อกำหนดดังต่อไปนี้ (40 คะแนน)

- 1.1. รายวิชา (Subject) ทั้งหมด included Course (รองรับ pagination และ sorting) (20 คะแนน)
- 1.2. รายชื่อ Course ตามรหัสวิชา (id) ที่ระบุ (10 คะแนน)
- 1.3. เพิ่ม (add) รายวิชาที่เปิดสอน (Courses) ต้องตรวจสอบไม่ให้มีการเปิดวิชาซ้ำในปีเดียวกัน (5 คะแนน)
- 1.4. ลบรายวิชาใน (CourseStudent) ของนักศึกษา ตามรหัสนักศึกษาและรหัสวิชา (id) ที่ระบุ (5 คะแนน)

*** การให้คะแนน ดูจาก โปรแกรมทำงานได้ถูกต้อง และ การออกแบบ ที่เหมาะสม ***

สิ่งที่ต้องส่ง (** Capture ส่ง ในรูปแบบไฟล์ PDF **)

- ผลลัพธ์ ของการทดสอบแต่ละ end-point (capture จาก postman)
- Prisma.schema / .env
- Source program (Router/Controller/Service/Data Access) เฉพาะ method/ส่วนของโปรแกรม ที่เกี่ยวกับ end-point นั้น ๆ ซึ่งแสดงให้เห็นว่า นักศึกษา เข้าใจ flow การทำงานของโปรแกรมที่เขียน
- ประเมินคะแนนเฉพาะข้อที่มีผลลัพธ์ส่งใน ไฟล์ PDF เท่านั้น
- Project (ในรูปแบบ zip) – excludes “node_modules” (ถ้ามี node_modules -2 คะแนน)

SQL Script

*** Database name: db_{your_student_id} ***

Create Tables:

```
create table courses(
    id          tinyint unsigned auto_increment primary key,
    subject_id  int    not null,
    year        year   not null,
    constraint courses_id_key unique (id)
);

create index idx_fk_subject_id
    on courses (subject_id);

create table students(
    id          tinyint unsigned auto_increment primary key,
    name        varchar(40) not null,
    constraint students_id_key unique (id)
);

create table course_student(
    course_id   tinyint unsigned not null,
    student_id  tinyint unsigned not null,
    grade       double           not null,
    primary key (course_id, student_id),
    constraint course_student_courses_id_fk
        foreign key (course_id) references courses (id),
    constraint course_student_students_id_fk
        foreign key (student_id) references students (id)
);

create index idx_fk_course_id
    on course_student (course_id);

create index idx_fk_student_id
    on course_student (student_id);

create table subjects(
    id          tinyint unsigned auto_increment primary key,
    subject_code char(6)           not null,
    title       varchar(40)       not null,
    credit      double default 3 not null,
    constraint subjects_id_key unique (id),
    constraint subjects_subject_code_key unique (subject_code)
);
```

Insert Data:

```
insert into subjects (subject_code, title, credit) values
    ('INT101', 'Computer Programming I', 1),
    ('INT202', 'Computer Programming 2', 2),
    ('INT303', 'Computer Programming 3', 3),
    ('INT404', 'Computer Programming 4', 4),
    ('INT105', 'Computer Programming 5', 5);
    ('INT206', 'Backend Dev 1', 2),
    ('INT307', 'Backend Dev 2', 3),
    ('INT408', 'Frontend Dev 1', 6),
    ('INT109', 'Frontend Dev 2', 3),
    ('INT210', 'Information Management 1', 3);
    ('INT310', 'Information Management 2', 3);
```

```
insert into courses (id, subject_id, year) values
    (1, 1, YEAR(NOW())-1),
    (2, 1, YEAR(NOW())),
    (3, 2, YEAR(NOW())),
    (4, 3, YEAR(NOW())),
    (5, 4, YEAR(NOW())),
    (6, 7, YEAR(NOW())),
    (7, 7, YEAR(NOW())-2),
    (8, 7, YEAR(NOW())-1),
    (9, 10, YEAR(NOW())-2),
    (10, 10, YEAR(NOW())-1),
    (11, 8, YEAR(NOW()));
```

```
insert into students values
    (101, 'Somchai'),
    (102, 'Apichart'),
    (103, 'Preeda'),
    (104, 'Kitichai'),
    (105, 'Satiya'),
    (106, 'Meechai'),
    (107, 'Srisupa'),
    (108, 'Daranee'),
    (109, 'Nittaya');
```

```
insert into course_student (course_id, student_id, grade) values
  (1, 101, 3.5),
  (1, 103, 2.5),
  (1, 104, 4.0),
  (2, 102, 2.5),
  (2, 105, 2.0),
  (3, 106, 2.5),
  (3, 107, 3.5),
  (3, 108, 1.5),
  (3, 109, 2.0),
  (4, 106, 2.5),
  (4, 107, 3.5),
  (5, 108, 1.5),
  (6, 101, 2.5),
  (6, 107, 3.5),
  (6, 102, 1.5),
  (5, 109, 2.0);
```
