# تقرير ربط نظام توزيع المراقبين بين Flutter, FastAPI, Laravel

هذا التقرير موجه لمطور Laravel الذي سيتولى ربط نظام توزيع المراقبين الذي يعتمد على Flutter كواجهة أمامية، وFastAPI لتنفيذ خوارزمية التوزيع، مع Laravel لحفظ النتائج النهائية في قاعدة البيانات MySQL.

## المكونات الأساسية

|  |  |  |
| --- | --- | --- |
| المكون | الوظيفة | منفذ الاتصال |
| Laravel | تخزين واسترجاع البيانات | 8001 و 8002 |
| FastAPI | تنفيذ توزيع المراقبين باستخدام OR-Tools | 8000 |
| Flutter | واجهة المستخدم | - |

## 1. ربط Flutter مع Laravel (جلب البيانات)

Flutter يرسل طلب GET إلى:  
GET http://127.0.0.1:8001/api/fetch-data  
  
يجب على Laravel إرجاع بيانات الجلسات والمراقبين بهذا الشكل:  
{  
 "sessions": [...],  
 "supervisors": [...]  
}  
  
ما يجب فعله:  
- إنشاء Route في Laravel: GET /api/fetch-data  
- إنشاء Controller يعيد قائمة الجلسات والمراقبين

## 2. FastAPI – توزيع المراقبين

FastAPI يتلقى بيانات من Flutter ويقوم بتوزيع المراقبين ويرجع جدولًا بالتوزيع النهائي.  
الواجهة الخاصة به جاهزة ولا تحتاج تعديل.

## 3. ربط Flutter مع Laravel (حفظ الجدول النهائي)

Flutter يرسل جدول التوزيع النهائي إلى:  
POST http://127.0.0.1:8002/api/save-schedule  
  
يجب أن يستقبل Laravel هذه البيانات ويخزنها في قاعدة البيانات.

### خطوات Laravel لحفظ البيانات

1. إعداد Route في routes/api.php:  
Route::post('/save-schedule', [ScheduleController::class, 'store']);  
  
2. إنشاء Model و Migration:  
php artisan make:model Assignment -m  
  
3. إضافة الأعمدة المناسبة في migration:  
- supervisor  
- day  
- time  
- room  
  
ثم تنفيذ:  
php artisan migrate  
  
4. إعداد ScheduleController:  
- استقبال البيانات  
- تخزين كل توزيع في جدول assignments

## ملاحظات إضافية

- تأكد من تفعيل CORS في Laravel.  
- تأكد أن Laravel و FastAPI يعملان على المنافذ المحددة.  
- يمكن اختبار API باستخدام Postman قبل ربطه بـ Flutter.