🟢 المشكلة قبل async/await

لما تعمل عمليات بطيئة (زي قراءة/كتابة على Database، ملفات، أو API calls) لو نفذتها في Main Thread ← البرنامج بيقف (يتجمد) لحد ما العملية تخلص.

مثال:

```
public IActionResult GetStudents()
{
    var students = context.Students.ToList(); // عملية بطيئة
    return View(students);
}
```

لو الداتا كتير، الويب أبليكشن ممكن يتأخر أو يهنّج.

🟢 الحل: async/await

async: بنحطها قدام الميثود عشان تقول للـ Compiler إن الميثود دي فيها عمليات هتتنفذ بشكل غير متزامن.

await: بنستخدمها قبل العملية البطيئة ← معناها: استنى النتيجة بس متوقفش الخيط (thread).

🟢 مثال عملي

```
public async Task<IActionResult> GetStudents()
{
    var students = await context.Students.ToListAsync(); // هنا await
    return View(students);
}
```

أثناء ما الاستعلام شغال، الـ Thread بيرجع للسيرفر يخدم Requests تانية.

أول ما العملية تخلص، يرجع يكمل الكود بعد await.

🟢 الفوائد

زيادة الأداء ← السيرفر يقدر يخدم مستخدمين أكتر في نفس الوقت.

تجربة مستخدم أفضل ← الواجهة (UI) ما بتتجمدش.

كود أوضح ← بدل الكولباك المعقدة، الكود يبقى شبه الـ Synchronous.

🟢 الخلاصة

async ← تخلي الميثود تشتغل بشكل غير متزامن.

await ← توقف تنفيذ السطر لحد ما المهمة تخلص من غير ما توقف الخيط.

النتيجة ← برامج أسرع، أكتر Responsive، وأسهل في القراءة والصيانة.