**Tesztelési Dokumentáció**

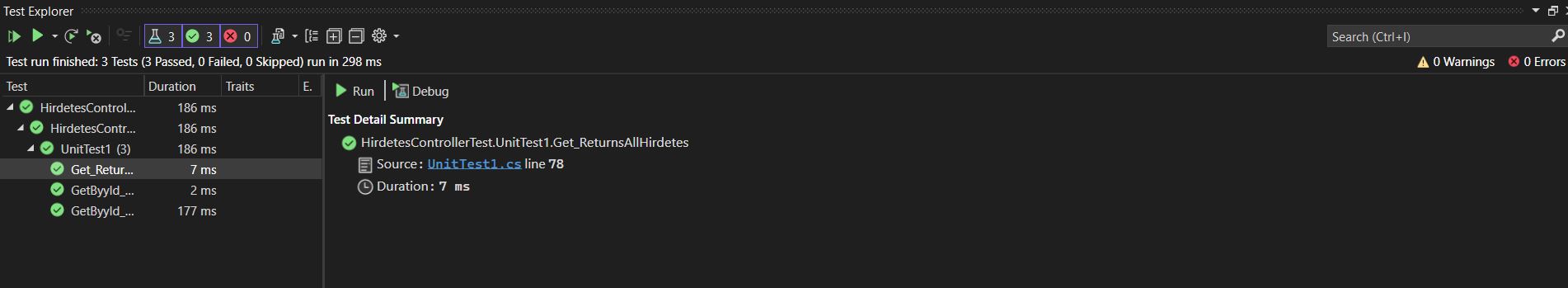
**Backend Tesztelése**

A tesztelés a HirdeteControllerben lévő Get és a GetByyId végpontokon történt. Asp.Net Web Api-n írt backend és a teszt keretrendszere pedig Moq és xUnit volt.

Get végpont tesztelése

Az elvárt az volt hogy a végpont az összes adatot vissza adja számunkra.

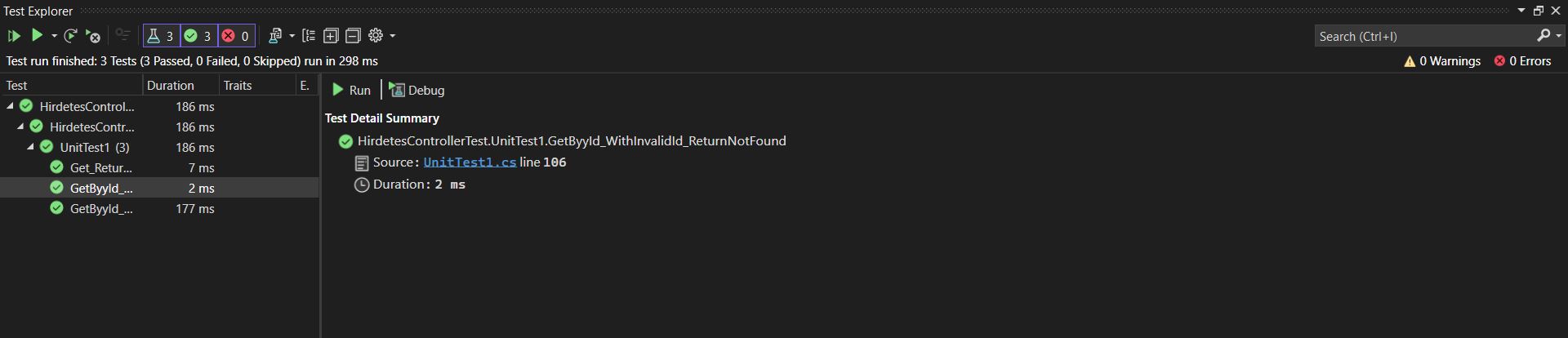
A teszt hiba nélkül lefutott és az elvárt adatokat vissza adta.



GetByyId végpont tesztelése. Helyes adatokkal.

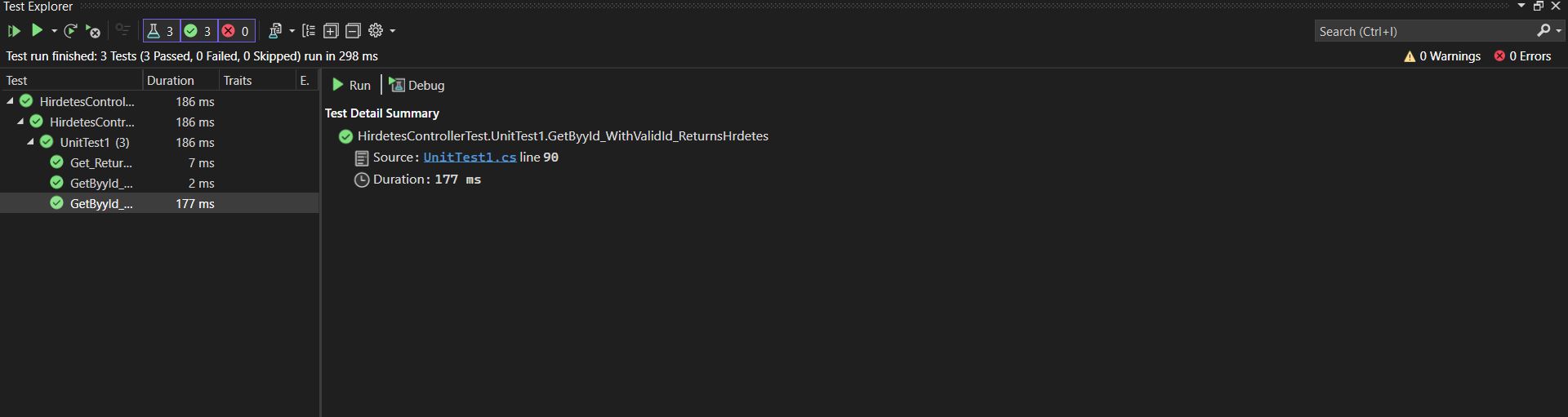
Az elvárt az volt hogy a kappot hirdetés Id-t és ha van ilyen azonosítóval rendelkező hirdetés akkor vissza adja az adott hirdetés összes adatát.

A teszt sikeresen lefutott vissza adva az adott hirdetés adatát.



GetByyId végpont tesztelése. Helytelen adatokkal.

Az elvárt az volt, hogy egy helytelen adat esetén NotFound-ot adjon visza.

A teszt sikeresen lefutott.

Frontend tesztelés:

**1. Tesztdokumentáció célja**

A dokumentáció célja, hogy biztosítsa a rendszer különböző funkcióinak helyes működését (pl. bejelentkezés, tokenkezelés, védett végpontok elérése stb.). A tesztelés célja a hibák és biztonsági hiányosságok azonosítása a hitelesítési és hozzáférési folyamatokban.

Egy egyszerű tesztdokumentációban a tesztesetek strukturált formában kerülnek bemutatásra. Az alábbiakban összefoglaljuk, hogyan nézhet ki egy tesztdokumentáció a leírt rendszerhez (ASP.NET Web API backend + React frontend, JWT alapú hitelesítés).

**4. Tesztesetek**

**A) Frontend tesztek**

**1. Bejelentkezési űrlap**

* **Teszteset**: Teszteljük az űrlap helyes működését.

| * **Lépés** | **Elvárt eredmény** |
| --- | --- |
|  |  |

H üres mező van akkor a küldés gomb lenyomására dobja ki, hogy „Töltse ki”

**Automatizált teszt (Jest + React Testing Library)**:

