

## 1. Klasör Yapısını Hazırla

Masaüstünde yeni bir klasör aç (Örn: Turnuva) ve dosyalarını şu düzenle içine yerleştir. **public klasörüne dikkat et, index.html onun içinde olmalı.**

Turnuva/

```
|--- public/  
|   |--- index.html (Arayüz kodları burada)  
|--- server.js (Sunucu kodları burada)  
|--- questions.js (Sorular burada)  
└--- package.json (Ayarlar burada)
```

## 2. Gerekli Paketleri Yükle

1. Bilgisayarında "Terminal'i veya "Komut İstemi"ni (CMD) aç.
2. Klasörünün içine gir (Örneğin: cd Desktop/Turuva).
3. Şu komutu yazıp Enter'a bas:

npm install

(Bu işlem node\_modules klasörünü oluşturacak ve gerekli kütüphaneleri indirecektir.)

## 3. Sunucuyu Başlat

Kurulum bittikten sonra aynı terminal ekranına şu komutu yaz:

node server.js

Ekranda "**Sunucu 3000 portunda hazır.**" yazısını gördüğünde her şey tamam demektir.

## 4. Oyuna Giriş Yap (Yerel Test)

- Tarayıcını aç ve adres çubuğuna `http://localhost:3000` yaz.
- **Öğretmen (Admin) Giriş:**
  - İsim: **admin**
  - Şifre: **-herhangi bir şifre-**
    - (*Bu girişle eşleştirme paneli ve oyuncu atma yetkileri açılır.*)
- **Öğrenci Giriş:**
  - İsim: (Herhangi bir isim, örn: Ali, Ayşe)

- (*Öğrenciler sadece bekleme ekranını görür.*)

## 5. Uzaktan Erişim (Ngrok ile Paylaşım)

Farklı bilgisayarlardan veya evlerden çocukların girmesi için:

1. node server.js çalışırken **yeni bir terminal penceresi daha aç**.
  2. Şu komutu yaz:
1. ngrok http 3000 Ngrok'un sana verdiği <https://....ngrok-free.app> linkini öğrencilere gönder.

Biz burada socket.io kullandık mı?

SOCKET.IO

Evet, **Socket.io** kütüphanesini bu dosyada (server.js) aktif olarak kullandık. Bu kütüphane, projenin "gerçek zamanlı" (real-time) çalışmasını sağlayan temel bileşendir.

Koddaki şu satırlar Socket.io kullanımını gösteriyor:

1. **Dahil Etme:** const { Server } = require("socket.io"); satırı ile kütüphaneyi çağırıyoruz.
2. **Kurulum:** const io = new Server(server); satırı ile Socket.io'yu HTTP sunucumuza bağlıyoruz.
3. **Haberleşme:** io.on('connection', ...) bloğu tüm bağlantıları yönetir. İçindeki socket.emit (veri gönderme) ve socket.on (veri alma) komutları sayesinde oyuncular ve sunucu birbirleriyle anlık olarak konuşur.

Bu sayede sayfayı yenilemeye gerek kalmadan sorular ekrana gelir ve cevaplar arasında sunucuya ulaşır.