



# ***“LEADERBORD”***

## ***FULL STACK UYGULAMA***

### ***DOKÜMANTASYONU***

Geliştirici: Beyazıt Bestami Sarıkaya

E-Posta: bestami980@outlook.com

Cep No: 0538 696 2189

Web Site: [bbestamis.com/](http://bbestamis.com/)

LinkedIn: [linkedin.com/in/beyazit-bestami-sarikaya-b016b8173](https://www.linkedin.com/in/beyazit-bestami-sarikaya-b016b8173)

Uygulama: <http://panteon.bbestamis.com/>

Proje Kodları: <https://github.com/BBestamiS/panteon>

V 1.0

09.11.2024, Ankara



## İÇİNDEKİLER

İÇİNDEKİLER.....	2
1. <i>GİRİŞ (INTRODUCTION)</i> .....	3
2. <i>MİMARİ (ARCHITECTURE)</i> .....	3
3. <i>BAŞLANGIÇ (GETTING STARTED)</i> .....	5
4. <i>DAĞITIM (DEPLOYMENT)</i> .....	8



## 1. GİRİŞ (INTRODUCTION)

**Uygulamanın Tanımı:** Bu proje, on milyon oyunculu bir oyun için verileri işleyip simüle eden ve bir merkezi sistemle entegre olan bir web uygulamasından oluşmaktadır. Projede kullanılan tüm teknolojiler, güvenli, hızlı ve ölçeklenebilir bir yapı sağlamak amacıyla seçilmiştir.

**Teknoloji Yığını (Tech Stack):** Projede kullanılan teknolojiler aşağıdaki gibidir.

- Front-End teknolojileri: React, Next.js, TypeScript, React-Icons, React Select, React Beautiful DnD
- Back-End teknolojileri: Node.js, Express.js, TypeScript, PM2, MySQL, Redis
- Database ve Cache teknolojileri: MySQL, Redis
- Server teknolojileri: Apache, Debian 12, Pm2, Byobu

## 2. MİMARİ (ARCHITECTURE)

### *Front-End*

Uygulama Adı: panteon-nextjs-app

Teknolojiler: TypeScript, Next.js, React, React-Icons, React Select, React Beautiful DnD

Açıklama:

"panteon-nextjs-app" uygulaması, TypeScript ve Next.js yapılarını kullanarak geliştirilmiş modern bir web uygulamasıdır. Kullanıcı arayüzü hızlı ve esnek olacak şekilde tasarlanmış olup, çeşitli React kütüphaneleri ile zenginleştirilmiştir. Kullanıcı deneyimini iyileştirmek için otomatik doldurma ve sürükle bırak özellikleri eklenmiştir.

### *Back-End*

Projenin back-end mimarisi iki ana uygulamadan oluşmaktadır:

1. panteon-ts-node-app
  - Amaç: "panteon-nextjs-app" ile merkezi sistem arasında veri akışını sağlayan bir API uygulamasıdır.
  - Teknolojiler: Node.js, Express.js, TypeScript, PM2
  - Açıklama: Bu uygulama, front-end ile back-end arasındaki iletişimi sağlayarak verilerin senkronizasyonunu üstlenir. Redis ile önbellekleme yapılır ve API uç noktaları üzerinden hızlı veri sunumu sağlanır.
2. panteon-ts-standalone-app
  - Amaç: On milyon oyuncuya sahip bir oyunun verilerini üretmek, işlemek ve sistem simülasyonu gerçekleştirmek.
  - Teknolojiler: Node.js, TypeScript
  - Açıklama: Bu uygulama, oyunun verilerini üretir, işler ve simüle eder. Veritabanı ve önbellekle sürekli etkileşimde bulunarak veri işleme hızını optimize eder.



### *Databases*

- MySQL: Relational Database olarak seçilmiştir ve projenin temel veri depolama kaynağıdır. Verilerin güvenli bir şekilde saklanmasını sağlar.
- Redis: Verilerin hızlı erişimi ve işlenmesi için cache görevi görmektedir. Bu sayede yüksek performans ve düşük gecikme sağlanmaktadır.

### *Server*

- Alan Adı: panteon.bbestamis.com
- Sunucu: AWS t3.medium
- İşletim Sistemi: Debian 12
- Teknolojiler: Apache, HaProxy, PM2, Byobu

#### Açıklama:

Uygulama, “panteon.bbestamis.com” alan adı üzerinden AWS sunucusuna yönlendirilmiştir. Debian 12 üzerinde çalışan sunucu, HaProxy ile yük dengeleme ve Apache ile web sunucu işlevlerini yerine getirmektedir. PM2, süreç yönetimi ve uygulamanın güvenilir şekilde çalışması için kullanılır.

#### Bağlantılar ve Veri Akışı

- Front-End ile Back-End Bağlantısı: “panteon-nextjs-app” uygulaması, “panteon-ts-node-app” API aracılığıyla merkezi sisteme bağlanır ve Redis önbellekle veri senkronizasyonunu sağlar.
- Back-End Veri İşleme ve Simülasyon: “panteon-ts-standalone-app” on milyon oyuncu için gerekli verileri üretir ve Redis ile işlenmiş verileri hızlıca sunar.
- Veritabanı ve Cache Etkileşimi: MySQL ana veri depolama alanıdır. Redis, sık kullanılan verilerin hızlı sunumu için cache işlevi görür.



### 3. BAŞLANGIÇ (GETTING STARTED)

**Ön Gereksinimler (Prerequisites):** Node.js, npm.js, MySQL, Redis, TypeScript,

**Kurulum (Installation):** Uygulamanın kurulumu için ilk adım veri tabanı kurulumlarının yapılmasıdır. Git bağlantısından erişim sağlayabileceğiniz “.sql” dosyaları üzerinden ilişkisel veri tabanı yapısını oluşturabilirsiniz.

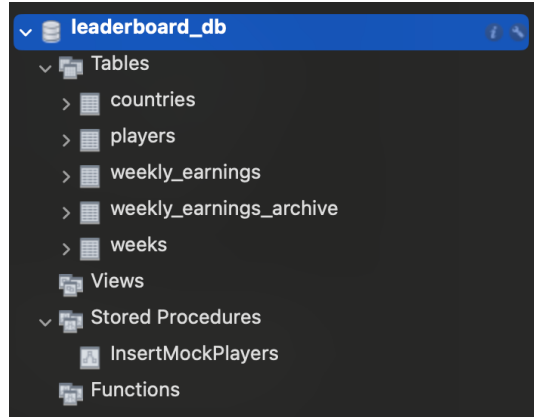
(Not: Tablolar ilişkilendirildiği için oluşturma sırası;

- `countries`,
- `weeks`,
- `players`,
- `weekly\_earnings`,
- `weekly\_earnings\_archive`,

Sırasında yapılmalıdır.)

- “leaderboard\_db\_countries.sql”,
- “leaderboard\_db\_countries\_mock.sql”,
- “leaderboard\_db\_players.sql”,
- “leaderboard\_db\_routines.sql”,
- “leaderboard\_db\_weekly\_earnings\_archive.sql”,
- “leaderboard\_db\_weekly\_earnings.sql”,
- “leaderboard\_db\_weeks.sql”

Gerekli kurulumları yaptıktan sonra veri tabanı yapısı;



bu şekilde görünecektir.

Öncelikle “leaderboard\_db\_countries\_mock.sql” dosyasının çalıştırılması gerekmektedir. Bu dosya içindeki kodlar ülke eklemesi yapacaktır.

On milyon mock player oluşturmak için “CALL InsertMockPlayers()” procedure’ünü çağırmanız gerekmektedir. Bu procedure’ün amacı sisteme on milyon birbirlerinden farklı/bağımsız kullanıcı yaratmaktır. (Not: İşlem uzun süreceğinden terminal üzerinden yapmanız önerilir.)



Veri tabanı kurulumu tamamlandıktan sonra back-end uygulamaları kurulur.

“**panteon-ts-node-app**” kurulumu/çalıştırılması için node.js ve npm.js’in sunucunuz içerisinde yer alması gerekmektedir.

Gerekli paketlerin kurulması gerekmektedir.

```
cd “project_path/”  
npm install
```

Komutlarıyla ilgili paketler sisteme yüklenir.

Proje dosyaları içerisinde yer alan “.env” dosyasının içerisine yer alan, bağlantı bilgilerin girilmesi gerekmektedir;

```
PORT=3000  
REDIS_HOST=127.0.0.1  
REDIS_PORT=6379
```

Uygulamayı çalıştırma scriptleri;

```
"scripts": {  
  "build": "tsc",  
  "start": "node ./dist/index.js",  
  "dev": "tsx watch src/index.ts",  
  "clean": "rm -rf dist",  
  "watch": "tsc -w"  
},
```

```
npm run build  
npm run start
```

Komutlarıyla API uygulamasını başlatabilirsiniz.

“**panteon-ts-standalone-app**” uygulaması sistem aktif oldukça sistemi simüle edecek olan uygulamadır. Bu sebeple durması durumunda kazançlar simüle edilemeyecektir.

Uygulamanın kurulumu/çalıştırılması için node.js ve npm.js’in sunucunuz içerisinde yer alması gerekmektedir.

Gerekli paketlerin kurulması gerekmektedir.

```
cd “project_path/”  
npm install
```

Komutlarıyla ilgili paketler sisteme yüklenir.



Proje dosyaları içerisinde yer alan ".env" dosyasının içerisine yer alan, bağlantı bilgilerin girilmesi gerekmektedir;

```
DB_HOST="****"  
DB_USER="****"  
DB_PASSWORD="****"  
DB_NAME="leaderboard_db"
```

```
REDIS_HOST="127.0.0.1"  
REDIS_PORT="6379"
```

Uygulamayı çalıştırma scriptleri;

```
"scripts": {  
  "build": "tsc",  
  "start": "node ./dist/index.js",  
  "dev": "tsx watch src/index.ts",  
  "clean": "rm -rf dist",  
  "watch": "tsc -w"  
},
```

```
npm run build  
npm run start
```

Back-end uygulamaları çalıştırıldıktan sonra Front-end Next.js uygulamasını çalıştırabilirsiniz.

**“panteon-nextjs-app”** front-end Next.js uygulamasıdır.

Uygulamanın kurulumu/çalışabilmesi için gerekli paketlerin kurulması gerekmektedir.

```
cd "project_path/"  
npm install
```

Komutlarıyla ilgili paketler sisteme yüklenir.

Uygulamayı çalıştırma scriptleri;

```
"scripts": {  
  "dev": "PORT=3001 next dev",  
  "build": "next build",  
  "start": "PORT=3001 next start",  
  "lint": "next lint"  
},
```

```
npm run build
```



npm run start

**Ortam Değişkenleri (Environment Variables):** “.env” dosyası içerisinde yer almaktadır. Kurulumlarda sisteme göre değiştirilmesi gerekmektedir.

“panteon-ts-standalone-app” .env dosya içeriği;

```
DB_HOST="****"  
DB_USER="****"  
DB_PASSWORD="****"  
DB_NAME="leaderboard_db"
```

```
REDIS_HOST="127.0.0.1"  
REDIS_PORT="6379"
```

“panteon-ts-node-app” .env dosya içeriği;

```
PORT=3000  
REDIS_HOST=127.0.0.1  
REDIS_PORT=6379
```

## 4. DAĞITIM (DEPLOYMENT)

**Dağıtım Ortamı:** AWS

**CI/CD Süreci:** Github (<https://github.com/BBestamiS/panteon>)