Dokumentacja Wdrożeniowa Systemu Quizyx

Zespół Deweloperski: Kamil Garus, Bartosz Mazur, Piotr Wodzisławski, Krzysztof Róg

22 stycznia 2026

Spis treści

1 Wstęp

Niniejsza dokumentacja opisuje szczegółowo proces instalacji, konfiguracji oraz uruchomienia aplikacji **Quizyx** (backend oraz frontend). Instrukcja obejmuje systemy operacyjne z rodziny Windows oraz Linux, a także zawiera wytyczne dotyczące wdrożenia produkcyjnego oraz debugowania.

2 Wymagania wstępne

Przed przystąpieniem do instalacji należy upewnić się, że środowisko docelowe spełnia poniższe wymagania:

- **Node.js**: Wersja zalecana LTS (np. 18.x lub 20.x).
- **Menedżer pakietów**: `npm` (domyślnie z Node.js) lub `yarn`.
- **Baza danych**: MongoDB (instancja lokalna lub zdalny serwer/Atlas).
- **System kontroli wersji**: Git (do sklonowania repozytorium). `pm2` (do zarządzania procesami w produkcji).

Zalecane zasoby sprzętowe: Minimum 1 GB RAM dla środowisk developerskich. W przypadku środowiska produkcyjnego zasoby należy skalować w zależności od przewidywanego obciążenia.

3 Konfiguracja repozytorium

3.1 Klonowanie projektu

Pierwszym krokiem jest pobranie kodu źródłowego i przejście do katalogu głównego projektu:

```
git clone <repo-url>
cd quizmaster-pro
```

3.2 Konfiguracja zmiennych środowiskowych

Należy utworzyć plik konfiguracyjny dla backendu o nazwie `.env` w katalogu `backend/`.

Przykładowa zawartość pliku `.env`:

```
JWT_SECRET=very_secret_jwt_key
MONGO_URI=mongodb://localhost:27017/quizmaster
PORT=5000
```

Uwaga: Ze względów bezpieczeństwa nie należy przechowywać wrażliwych danych (haseł, kluczy API) w systemie kontroli wersji.

4 Uruchomienie w trybie developerskim

Poniższe instrukcje mają zastosowanie zarówno dla systemów Windows, jak i Linux.

4.1 Backend (API)

Uruchomienie serwera backendowego w trybie nasłuchiwania na zmiany (hot-reload):

```
cd backend
npm install

# Kopiowanie pliku konfiguracyjnego (jesli dostepny jest przyklad)
# Windows:
copy .env.example .env
# Linux/macOS:
cp .env.example .env

# Uruchomienie trybu developerskiego (nodemon + ts-node)
npm run dev
```

Po poprawnym uruchomieniu w konsoli powinny pojawić się komunikaty: *"MongoDB Connected"* oraz informacja o porcie serwera (np. *"Server running on port 5000"*).

4.2 Frontend (Klient)

Uruchomienie aplikacji klienckiej:

```
cd ../frontend
npm install
npm run dev
```

Serwer deweloperski Vite udostępni aplikację pod adresem lokalnym, zazwyczaj `http://localhost:517`

5 Uruchomienie produkcyjne

5.1 Frontend (Build statyczny)

Dla środowiska produkcyjnego aplikacja frontendowa powinna zostać zbudowana do plików statycznych:

```
cd frontend
npm run build
```

Wynikiem operacji jest katalog `dist/`, który jest gotowy do serwowania przez serwer WWW (np. Nginx, Apache, Caddy).

5.2 Backend

W środowisku produkcyjnym zaleca się skompilowanie kodu TypeScript do JavaScript lub użycie menedżera procesów.

5.2.1 Metoda 1: Kompilacja (Zalecane)

```
cd backend
npm install
npx tsc
# Uruchomienie skompilowanego pliku
node dist/server.js
```

5.2.2 Metoda 2: PM2 (Process Manager)

Uruchomienie bez wstępnej kompilacji przy użyciu `ts-node`:

```
pm2 start "node -r ts-node/register server.ts" --name quizmaster-backend
pm2 save
```

6 Zmienne środowiskowe

Pełna lista zmiennych konfiguracyjnych:

Zmienna	Opis	Wymagalność
`JWT_SECRET`	Klucz sekretny do podpisywania tokenów JWT	Wymagane
`MONGO_URI`	Adres połączenia do bazy MongoDB	Wymagane
`PORT`	Port nasłuchiwania backendu (domyślnie 5000)	Opcjonalne

Tabela 1: Zmienne środowiskowe backendu

7 Konfiguracja sieci i Reverse Proxy

W środowisku produkcyjnym zaleca się użycie Nginx jako reverse-proxy oraz do terminacji połączeń HTTPS.

Przykładowa konfiguracja bloku `server` w Nginx:

```
server {
    listen 80;
    server_name example.com;

    # Proxy dla API
    location /api/ {
        proxy_pass http://127.0.0.1:5000/api/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    # Serwowanie Frontendu (SPA)
    location / {
        root /var/www/frontend/dist;
        try_files $uri $uri/ /index.html;
    }
}
```

8 Konserwacja i Monitoring

8.1 Backup danych

Zaleca się regularne tworzenie kopii zapasowych bazy danych. W przypadku wdrożenia prostego (Docker) można backupować wolumen `mongo-data`. Przy korzystaniu z MongoDB Atlas należy skonfigurować automatyczne snapshoty.

8.2 Monitoring i logi

Do monitorowania dostępności usługi w produkcji należy wykorzystywać `pm2` lub systemowy `systemd`. Logi powinny podlegać rotacji, aby nie zapełnić przestrzeni dyskowej serwera.

9 Rozwiązywanie problemów (Troubleshooting)

- **Błąd `ECONNREFUSED` (MongoDB)**: Sprawdź poprawność `MONGO_URI` oraz czy proces bazy danych jest aktywny.
- **Błąd CORS**: Upewnij się, że backend ma skonfigurowaną politykę CORS zezwalającą na zapytania z domeny frontendu.
- **Socket.io nie łączy**: Zweryfikuj, czy reverse-proxy (Nginx) poprawnie obsługuje nagłówki WebSocket (Upgrade/Connection) oraz czy porty nie są blokowane przez firewall.