



Fundusze Europejskie
dla Nowoczesnej Gospodarki



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Określenie przedmiotu zamówienia

Przedmiotem zamówienia jest: **usługa zaprojektowania, opracowania, implementacji, wdrożenia i utrzymania oprogramowania systemu szkoleniowego AR Bridge Simulator z wykorzystaniem rzeczywistości rozszerzonej, składającego się z trzech komponentów aplikacji AR/MR na gogle wirtualnej rzeczywistości Meta Quest 3, serwera backendowego oraz aplikacji przeglądarkowej (panel nadzorcy).**

System ma umożliwiać:

- prowadzenie egzaminów i szkoleń z wykorzystaniem AR,
- tworzenie i realizacją scenariuszy szkoleniowych,
- automatyczną ocenę działań kursanta,
- monitorowanie przebiegu szkolenia w czasie rzeczywistym.

Scenariusze (3 szt.) zostaną dostarczone przez Zamawiającego.

System składa się z trzech głównych komponentów:

1. Aplikacja AR/MR (gogle Meta Quest 3),
2. Serwer backendowy,
3. Aplikacja przeglądarkowa (panel nadzorcy).

Zgodnie ze specyfikacją techniczną system musi obsługiwać m.in. telemetrię, streaming wideo, komunikację w czasie rzeczywistym oraz zarządzanie scenariuszami. Interface ma zostać wykonany w j. angielskim.

Zakres prac:

- zaprojektowanie architektury systemu
- implementacja aplikacji AR (Unity, Meta Quest 3)
- implementacja serwera (.NET, SignalR, PostgreSQL)
- implementacja aplikacji przeglądarkowej (React)
- integracja komponentów systemu
- wdrożenie środowisk (staging+ produkcja)
- przeprowadzenie testów akceptacyjnych
- przygotowanie dokumentacji technicznej i użytkowej
- świadczenie usług utrzymania przez okres 24 miesięcy z raportem utrzymaniowym na żądanie Zamawiającego

Zamawiający oczekuje, że w ramach usługi oprogramowania:

- aplikacja AR łączy się z serwerem i wysyła telemetrię
- serwer odbiera telemetrię i rozgłasza ją do panelu nadzorcy via SignalR
- panel nadzorcy wyświetla listę urządzeń ze statusem i dane telemetryczne
- daje możliwość zdalnego uruchomienia i zatrzymania scenariusza z panelu nadzorcy
- działa pipeline CI/CD z wdrożeniem na staging
- wszystkie wymagania oznaczone w specyfikacji technicznej (załącznik 2 do umowy) jako "wymagane" są zaimplementowane i działają poprawnie
- transmisja sceny AR z gogli wirtualnej rzeczywistości Meta Quest 3 działa z opóźnieniem poniżej 2 sekund
- scenariusze szkoleniowe z automatycznymi kryteriami oceny działają na symulatorze mostka nawigacyjnego
- historia sesji i ocen jest zapisywana w bazie danych i dostępna z panelu nadzorcy



Fundusze Europejskie
dla Nowoczesnej Gospodarki



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



- system zarządzania scenariuszami i zestawami sesji działa poprawnie
- uwierzytelnianie i autoryzacja działają poprawnie
- dokumentacja jest kompletna i przekazana
- procedura przywracania z kopii zapasowej została przetestowana
- testy akceptacyjne przeprowadzone z udziałem Zamawiającego zakończone pozytywnie

Kluczowe funkcjonalności systemu m.in.:

- tworzenie i uruchamianie scenariuszy szLoleniowych w AR,
- skanowanie i odwzorowanie przestrzeni mostka nawigacyjnego,
- nakładanie obiektów AR na rzeczywiste środowisko,
- zbieranie i analizę danych telemetrycznych,
- streaming obrazu z gogli w czasie rzeczywistym,
- zdalne sterowanie scenariuszami przez egzaminatora,
- automatyczną ocenę działań kursanta,
- zarządzanie sesjami szkoleniowymi i historią wyników.

Wymagania technologiczne (wysokopoziomowe)

Preferowany stack technologiczny obejmuje:

- Unity + C# (aplikacja AR),
- .NET + SignalR (backend),
- PostgreSQL (baza danych),
- React + TypeScript (panel webowy),
- Docker (wdrożenie),
- komunikacja REST, WebSocket, WebRTC.

Dopuszcza się alternatywne rozwiązania technologiczne po uzasadnieniu.

Wymagania niefunkcjonalne (wybrane)

- opóźnienie transmisji obrazu ≤ 2 s,
- opóźnienie telemetry ≤ 500 ms,
- dostępność systemu * 99%,
- bezpieczeństwo komunikacji (HTTPS/WSS),
- autoryzacja użytkowników (JWT) .

Utrzymanie

- świadczenia usług utrzymaniowych przez 24 miesiące,
- usuwania błędów i aktualizacji systemu,
- zapewnienia SLA zgodnie z poziomami krytyczności.

Prace będą wykonywane w terminie maksymalnie 17 tygodni, jednak nie dłużej niż do 31.01.2027 r. w podziale na dwa etapy:

Realizacja przedmiotu zamówienia	Zakres funkcjonalno-techniczny – kryteria odbioru oraz wymagania dot. dokumentacji
Etap I	<p>Opracowanie prototypu aplikacji (MVP)</p> <ol style="list-style-type: none"> 1. Aplikacja AR łączy się z serwerem i wysyła telemetry 2. Serwer odbiera telemetry i rozgłasza ją do panelu nadzorcy via SignalR 3. Panel nadzorcy wyświetla listę urządzeń ze statusem i dane telemetryczne 4. Możliwość zdalnego uruchomienia i zatrzymania scenariusza z panelu nadzorcy



	<p>5. Działający pipeline CI/CD z wdrożeniem na staging</p> <p>Wymagane przekazanie: DOC-001: Dokumentacja techniczna architektury systemu (diagramy, opis komponentów, przepływ danych) DOC-002: Dokumentacja API serwera (OpenAPI/Swagger)</p>
<p>Etap II</p>	<p>Implementacja przetestowanego oprogramowania, opracowanie dokumentacji technicznej oraz kodu źródłowego, dostępu</p> <ol style="list-style-type: none"> 1. Wszystkie wymagania oznaczone jako „Wymagane” są zimplementowane i działają poprawnie 2. Transmisja sceny AR z gogli wirtualnej rzeczywistości Meta Quest 3 działa z opóźnieniem poniżej 2 sekund 3. Scenariusze szkoleniowe z automatycznymi kryteriami oceny działają na symulatorze mostka nawigacyjnego 4. Historia sesji i ocen jest zapisywana w bazie danych i dostępna z panelu nadzorczy 5. System zarządzania scenariuszami i zestawami sesji działa poprawnie 6. Uwierzytelnianie i autoryzacja działają poprawnie 7. Dokumentacja jest kompletna i przekazana 8. Procedura przywracania z kopii zapasowej została przetestowana 9. Testy akceptacyjne przeprowadzone z udziałem Zamawiającego zakończone pozytywnie <p>Wymagane przekazanie: DOC-003: Instrukcja wdrożenia i konfiguracji systemu (serwer, aplikacja przeglądarkowa, aplikacja AR) DOC-004: Instrukcja użytkownika panelu nadzorczy DOC-005: Instrukcja instalacji i konfiguracji aplikacji AR na gogle wirtualnej rzeczywistości Meta Quest 3 DOC – 006: Procedura przywracania systemu z kopii zapasowej</p> <p>Kodu źródłowy opracowanej aplikacji, kody dostępowe, hasła</p>

Przedmiot umowy realizowany jest w ramach zadania nr 1 „Inkubator Rozwoju” projektu niekonkurencyjnego pn. „Science4Business - Nauka dla Biznesu” na podstawie Porozumienia nr FENG.02.05-IP.07-001/23 z dnia 17 kwietnia 2025 r. oraz w ramach umowy nr MNiSW/2025/DPI/251 o partnerstwie przy wspólnej realizacji projektu, w ramach Programu Fundusze Europejskie dla Nowoczesnej Gospodarki 2021-2027 współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego, Działanie FENG.02.05.

Warunki realizacji zamówienia:

Przedmiot zamówienia powinien zostać zrealizowany w terminie nie dłuższym niż 17 tygodni, zgodnie z umową o wykonanie usługi wskazaną w załączeniu. W ramach umowy wymagane jest przeniesienie autorskich praw majątkowych i praw własności przemysłowej na UMG za wynagrodzeniem i zachowanie informacji poufnych w poufności.

Warunki płatności:

Zgodnie z warunkami umowy o wykonanie usługi, stanowiącej wzór do postępowania.

Wynagrodzenie wypłacane w dwóch transzach:

- pierwsza transza po zakończeniu I etapu prac
- druga transza po zakończeniu II etapu prac

Każda z transz po 50% wartości zamówienia.



Specyfikacja techniczna:

Specyfikacja wymagań technicznych dla oprogramowania			
1. Architektura systemu			
1.1. Diagram architektury			
Komponent	Wymagana technologia	Propozycje szczegółów implementacyjnych*	Komunikacja
Aplikacja AR/MR	Unity, C#	Unity 6 LTS, Universal Render Pipeline, Meta XR All-in-One SDK	↔ Serwer (REST, WebSocket, WebRTC)
Serwer	.NET, SignalR	.NET 9, Minimal API, Entity Framework Core	↔ Aplikacja AR, ↔ Aplikacja przeglądarkowa
Baza danych	PostgreSQL	PostgreSQL 16	+ ← Serwer
Aplikacja przeglądarkowa	React, TypeScript	React 19, Vite, Tailwind CSS	↔ Serwer (SignalR/WebSocket)
Infrastruktura	Docker, platforma hostingowa	—	Serwer + Aplikacja przeglądarkowa + Baza danych
* Kolumna zawiera rekomendowane wersje i biblioteki. Wykonawca może zaproponować alternatywne wersje lub biblioteki, o ile będzie to uzasadnione korzyścią dla projektu i zaakceptowane przez Zamawiającego.			
1.2 Zasady komunikacji			
<ul style="list-style-type: none"> i. Aplikacja AR komunikuje się wyłącznie z serwerem — nigdy bezpośrednio z panelem nadzorcy ii. Cała komunikacja aplikacja przeglądarkowa ↔ AR jest przekazywana za pośrednictwem serwera iii. Komunikacja jest dwukierunkowa: iv. AR → Serwer → Aplikacja przeglądarkowa: telemetria, status scenariusza, transmisja sceny AR, postęp kryteriów v. Aplikacja przeglądarkowa → Serwer → AR: uruchomienie/zatrzymanie scenariusza, komendy konfiguracyjne 			
1.3. Protokoły komunikacji			
Kierunek	Protokół	Zastosowanie	
AR → Serwer	REST (HTTP POST)	Telemetria, status urządzenia	
AR ↔ Serwer	SignalR (WebSocket)	Komendy dwukierunkowe, sterowanie scenariuszami	
AR → Serwer	WebRTC / RTSP	Transmisja sceny AR z gogli	
Serwer ↔ Aplikacja przeglądarkowa	SignalR (WebSocket)	Dane w czasie rzeczywistym, komendy	
2. Wymagania funkcjonalne			
2.1. Aplikacja AR/MR (Unity — gogle wirtualnej rzeczywistości Meta Quest 3)			
2.1.1. Skanowanie otoczenia i interakcja AR			
ID	Wymaganie	Priorytet	



AR-001	Aplikacja musi wykorzystywać funkcję Passthrough gogli wirtualnej rzeczywistości Meta Quest 3 do wyświetlania rzeczywistości rozszerzonej (mixed reality)	Wymagane
AR-002	Aplikacja musi umożliwiać skanowanie otoczenia symulatora mostka nawigacyjnego z wykorzystaniem Meta Scene API (Scene Mesh, Spatial Anchors)	Wymagane
AR-003	System musi rozpoznawać kluczowe elementy mostka nawigacyjnego na podstawie zeskanowanej geometrii lub predefiniowanych kotwic przestrzennych (Spatial Anchors)	Wymagane
AR-004	Aplikacja musi umożliwiać nakładanie wirtualnych obiektów AR na rzeczywiste otoczenie symulatora w celu kreowania scenariuszy szkoleniowych (np. symulacja ognia, rozlania płynów, alarmy wizualne)	Wymagane
AR-005	Interakcja z obiektami AR musi być możliwa za pomocą śledzenia dłoni (Hand Tracking) jako podstawowej metody interakcji	Wymagane
3.1.2. System scenariuszy szkoleniowych		
ID	Wymaganie	Priorytet
AR-010	Aplikacja musi obsługiwać zdalne uruchamianie scenariuszy szkoleniowych — komenda startu przychodzi z serwera (zainicjowana przez egzaminatora z poziomu panelu nadzorca)	Wymagane
AR-011	Aplikacja musi obsługiwać zdalne zatrzymywanie aktywnego scenariusza	Wymagane
AR-012	Każdy scenariusz musi posiadać zdefiniowaną listę kryteriów oceny, które są automatycznie monitorowane i raportowane do serwera w czasie rzeczywistym	Wymagane
AR-013	Kryteria oceny muszą być weryfikowane automatycznie na podstawie akcji kursanta w środowisku AR (np. podejście do gaśnicy, użycie wyposażenia, komunikacja)	Wymagane
AR-014	Aplikacja musi raportować do serwera moment spełnienia każdego kryterium wraz ze znacznikiem czasu	Wymagane
AR-015	Aplikacja musi obsługiwać kategorie scenariuszy: bezpieczeństwo (safety), komunikacja (communication), reakcja (reaction) oraz inne (others)	Wymagane
AR-016	Aplikacja musi umożliwiać pracę w trybie offline z synchronizacją wyników po przywróceniu połączenia	Wymagane
1.1.3. Telemetria i komunikacja z serwerem		
ID	Wymaganie	Priorytet
AR-020	Aplikacja musi wysyłać dane telemetryczne do serwera z konfigurowalną częstotliwością	Wymagane



AR-021	Dane telemetryczne muszą zawierać co minimum: identyfikator urządzenia, pozycję (x, y, z), stan baterii, jakość połączenia, aktywny scenariusz, postęp kryteriów oceny, znacznik czasu	Wymagane
AR-023	Aplikacja musi utrzymywać dwukierunkowe połączenie z serwerem (SignalR/WebSocket) do odbioru komend	Wymagane
AR-024	Aplikacja musi implementować mechanizm automatycznego ponownego łączenia z serwerem w przypadku utraty połączenia	Wymagane
3.1.4. Transmisja sceny AR (AR video streaming)		
ID	Wymaganie	Priorytet
AR-030	Aplikacja musi umożliwiać strumieniowanie renderowanej sceny AR (wirtualnych elementów nałożonych na otoczenie) do serwera	Wymagane
AR-031	Transmisja obejmuje obraz generowany przez aplikację AR — wirtualne obiekty renderowane przez silnik Unity, bez obrazu z kamery passthrough	Wymagane
AR-032	Opóźnienie transmisji sceny AR nie powinno przekraczać 2 sekund (end-to-end)	Wymagane
AR-033	Rozdzielczość transmisji sceny AR musi być wystarczająca do monitorowania działań kursanta przez egzaminatora	Wymagane
AR-034	Transmisja musi być optymalizowana pod kątem przepustowości sieci Wi-Fi centrum szkoleniowego	Wymagane
1.1.5. Wymagania techniczne aplikacji AR		
ID	Wymaganie	Priorytet
AR-040	Silnik: Unity (sugerowana wersja: 6 LTS)	Wymagane
AR-041	Język programowania: C#	Wymagane
AR-042	Platforma docelowa: gogle wirtualnej rzeczywistości Meta Quest 3	Wymagane
AR-043	Aplikacja musi utrzymywać stabilną liczbę klatek na sekundę zapewniającą płynne działanie na goglach wirtualnej rzeczywistości Meta Quest 3	Wymagane
AR-044	Konfiguracja serwera (URL, identyfikator urządzenia, interwał telemetrii) musi być zarządzana przez plik konfiguracyjny	Wymagane
1.2. Serwer (.NET)		
1.2.1. API i komunikacja		
ID	Wymaganie	Priorytet
SRV-001	Serwer musi być zbudowany w architekturze .NET (sugerowana wersja: 9) z wykorzystaniem Minimal API	Wymagane
SRV-002	Serwer musi udostępniać hub SignalR do komunikacji w czasie rzeczywistym z aplikacją przeglądarkową	Wymagane
SRV-003	Serwer musi przyjmować dane telemetryczne od aplikacji AR poprzez REST API i rozgłaszać je do podłączonych klientów SignalR	Wymagane



SRV-004	Serwer musi utrzymywać dwukierunkowe połączenie SignalR/WebSocket z aplikacją AR do przekazywania komend (start/stop scenariusza, konfiguracja)	Wymagane
SRV-005	Serwer musi udostępniać endpoint health check (GET /api/health)	Wymagane
SRV-006	Serwer musi walidować dane wejściowe (telemetrię, komendy) i zwracać odpowiednie kody błędów	Wymagane
SRV-007	Serwer musi obsługiwać przekazywanie strumienia sceny AR z aplikacji AR do aplikacji przeglądarkowej (relay/proxy lub sygnalizacja WebRTC)	Wymagane
1.2.2. Baza danych i persystencja		
ID	Wymaganie	Priorytet
SRV-010	System musi wykorzystywać relacyjną bazę danych (PostgreSQL) do trwałego przechowywania danych	Wymagane
SRV-011	Baza danych musi przechowywać pełną historię telemetrii z sesji szkoleniowych	Wymagane
SRV-012	Baza danych musi przechowywać wyniki ocen scenariuszy szkoleniowych: które kryteria zostały spełnione, czasy realizacji, wynik końcowy	Wymagane
SRV-013	Baza danych musi przechowywać definicje scenariuszy szkoleniowych i ich kryteriów oceny	Wymagane
SRV-014	Baza danych musi przechowywać definicje zestawów scenariuszy (sesji szkoleniowych)	Wymagane
SRV-015	Baza danych musi przechowywać dane użytkowników (egzaminatorów) i urzędzeń	Wymagane
SRV-016	Serwer musi wykorzystywać ORM do dostępu do bazy danych (sugerowany: Entity Framework Core)	Wymagane
SRV-017	Migracje schematu bazy danych muszą być zarządzane przez mechanizm migracji wybranego ORM	Wymagane
3.2.3. Zarządzanie scenariuszami		
ID	Wymaganie	Priorytet
SRV-020	Scenariusze szkoleniowe i ich kryteria oceny są predefiniowane — serwer musi udostępniać REST API do odczytu scenariuszy i ich kryteriów	Wymagane
SRV-021	Serwer musi udostępniać REST API do tworzenia zestawów scenariuszy (sesji szkoleniowych) — grupowania wybranych predefiniowanych scenariuszy w uporządkowaną sekwencję na daną sesję	Wymagane
SRV-023	Serwer musi udostępniać REST API do pobierania historii przeprowadzonych sesji szkoleniowych wraz z wynikami	Wymagane
SRV-024	Serwer musi obsługiwać przypisywanie zestawu scenariuszy do konkretnego urządzenia/sesji	Wymagane



1.2.4. Zarządzanie urządzeniami		
ID	Wymaganie	Priorytet
SRV-030	Serwer musi przechowywać rejestr urządzeń (gogli) z ich identyfikatorami i nazwami	Wymagane
SRV-031	Serwer musi śledzić status połączenia każdego urządzenia w czasie rzeczywistym	Wymagane
SRV-032	Serwer musi przechowywać i udostępniać ostatni znany stan urządzenia (bateria, aktywny scenariusz)	Wymagane
3.3. Panel nadzorca (aplikacja przeglądarkowa — React)		
3.3.1. Połączenie i monitorowanie		
ID	Wymaganie	Priorytet
WEB-001	Panel musi umożliwiać nawiązanie połączenia z serwerem za pośrednictwem SignalR	Wymagane
WEB-002	Panel musi wyświetlać listę zarejestrowanych urządzeń (gogli) z ich statusem: nazwa, poziom baterii, status połączenia (online/offline)	Wymagane
WEB-003	Panel musi automatycznie aktualizować status urządzeń w czasie rzeczywistym	Wymagane
WEB-004	Panel musi obsługiwać automatyczne ponowne połączenie z serwerem	Wymagane
WEB-005	Panel musi wyświetlać log zdarzeń systemowych (połączenia, błędy, zdarzenia scenariuszy) z oznaczeniem czasu i typu (info, warning, error, success)	Wymagane
1.3.2. Transmisja sceny AR		
ID	Wymaganie	Priorytet
WEB-010	Panel musi wyświetlać transmisję sceny AR na żywo z wybranego urządzenia (gogli wirtualnej rzeczywistości Meta Quest 3)	Wymagane
WEB-011	Transmisja musi pokazywać renderowaną scenę AR — wirtualne obiekty generowane przez aplikację	Wymagane
WEB-012	Panel musi umożliwiać przełączanie podglądu między różnymi urządzeniami	Wymagane
1.3.3. Zarządzanie scenariuszami i sesjami		
ID	Wymaganie	Priorytet
WEB-020	Panel musi umożliwiać zdalne uruchomienie wybranego scenariusza na wskazanym urządzeniu	Wymagane
WEB-021	Panel musi umożliwiać zdalne zatrzymanie aktywnego scenariusza	Wymagane
WEB-022	Panel musi wyświetlać scenariusze pogrupowane według kategorii z możliwością rozwijania/zwijania kategorii	Wymagane
WEB-023	Panel musi wyświetlać w czasie rzeczywistym postęp realizacji kryteriów aktywnego scenariusza (checklist z automatyczną aktualizacją)	Wymagane
WEB-024	Panel musi wyświetlać czas trwania aktywnego scenariusza	Wymagane



WEB-025	Panel musi wyświetlać pasek postępu pokazujący procent spełnionych kryteriów	Wymagane
WEB-026	Panel musi umożliwiać tworzenie i edycję zestawów scenariuszy na sesję szkoleniową (wybór scenariuszy, kolejność)	Wymagane
WEB-027	Panel musi umożliwiać przeglądanie historii przeprowadzonych sesji z wynikami ocen	Wymagane
WEB-028	Panel musi umożliwiać przeglądanie predefiniowanych scenariuszy szkoleniowych oraz ich kryteriów oceny	Wymagane
1.3.4. Notatki i raportowanie		
ID	Wymaganie	Priorytet
WEB-030	Panel musi umożliwiać prowadzenie notatek tekstowych per urządzenie/sesja	Wymagane
WEB-031	Panel musi umożliwiać eksport wyników sesji szkoleniowej (format eksportu do ustalenia)	Wymagane
ID	Wymaganie	Priorytet
WEB-040	Framework: React z TypeScript (sugerowana wersja: React 19)	Wymagane
WEB-041	Bundler (sugerowany: Vite)	Wymagane
WEB-042	Framework CSS (sugerowany: Tailwind CSS)	Wymagane
WEB-043	Komunikacja z serwerem: klient SignalR	Wymagane
WEB-044	Panel musi być responsywny i działać poprawnie na ekranach o rozdzielczości od 1280x720 wwyż	Wymagane
WEB-045	Panel musi działać poprawnie w przeglądarkach: Chrome (ostatnie 2 wersje), Firefox (ostatnie 2 wersje), Edge (ostatnie 2 wersje)	Wymagane
2. Wymagania нефункционалне		
ID	Wymaganie	Priorytet
NF-001	Opóźnienie transmisji telemetrii (AR → aplikacja przeglądarkowa) nie powinno przekraczać 500ms	Wymagane
NF-002	Opóźnienie transmisji sceny AR (AR → aplikacja przeglądarkowa) nie powinno przekraczać 2 sekund	Wymagane
NF-003	Dostępność systemu (serwer + aplikacja przeglądarkowa) w środowisku produkcyjnym: minimum 99% w skali miesiąca	Wymagane
3. Wymagania technologiczne		
3.1. Wymagany stack technologiczny		
Komponent	Wymagana technologia	Propozycja wersji / szczegółów
Aplikacja AR	Unity	6 LTS (6000.x)
Aplikacja AR	C#	Zgodna z wybraną wersją Unity
Aplikacja AR	Pipeline renderowania (URP lub inny)	Universal Render Pipeline
Aplikacja AR	SDK wirtualnej rzeczywistości Meta	Meta XR All-in-One SDK, najnowsza stabilna
Serwer	.NET	9.0



Serwer	SignalR	Zgodny z wybraną wersją .NET
Serwer	ORM	Entity Framework Core 9.0
Serwer	PostgreSQL	16+
Aplikacja przeglądarkowa	React	19.x
Aplikacja przeglądarkowa	TypeScript	5.x
Aplikacja przeglądarkowa	Bundler	Vite 6.x
Aplikacja przeglądarkowa	Framework CSS	Tailwind CSS 3.x
Aplikacja przeglądarkowa	Klient SignalR	@microsoft/signalr 10.x
Infrastruktura	Docker	Najnowsza stabilna
Platforma docelowa AR	Meta Quest 3	HorizonOS SDK 60+
<i>* Podane wersje i szczegółowe biblioteki stanowią propozycję Zamawiającego. Wykonawca może zaproponować alternatywne rozwiązania, o ile będzie to uzasadnione korzyścią dla projektu i zaakceptowane przez Zamawiającego.</i>		
4. Bezpieczeństwo		
4.1. Autoryzacja i uwierzytelnianie		
ID	Wymaganie	Priorytet
SEC-001	Panel nadzorcy musi wymagać uwierzytelnienia użytkownika (login + hasło)	Wymagane
SEC-002	System musi implementować autoryzację opartą o tokeny (JWT)	Wymagane
SEC-003	Konta użytkowników (egzaminatorów) są prekonfigurowane — system nie wymaga rejestracji	Wymagane
SEC-004	Hasła muszą być przechowywane w postaci zahasłowanej	Wymagane
SEC-005	Sesje JWT muszą mieć konfigurowalny czas wygaśnięcia i obsługiwać mechanizm odświeżania tokenu	Wymagane
4.2. Szyfrowanie i bezpieczeństwo komunikacji		
ID	Wymaganie	Priorytet
SEC-010	Komunikacja między panelem nadzorcy a serwerem musi być szyfrowana (HTTPS/WSS)	Wymagane
SEC-011	Komunikacja między aplikacją AR a serwerem musi być szyfrowana (HTTPS/WSS) w środowisku produkcyjnym	Wymagane
SEC-012	Połączenia SignalR muszą być uwierzytelniane — nieautoryzowani klienci nie mogą subskrybować danych	Wymagane
SEC-013	Urządzenia AR muszą uwierzytelniać się na serwerze za pomocą klucza API lub certyfikatu	Wymagane
5. Infrastruktura i DevOps		
5.1. Środowiska		
ID	Wymaganie	Priorytet
INF-001	System musi posiadać co najmniej dwa środowiska: developerskie (staging) i produkcyjne	Wymagane
INF-002	Środowisko produkcyjne hostowane na platformie hostingowej zapewniającej	Wymagane



	bezpieczeństwo i stabilność działania (do ustalenia z Wykonawcą	
INF-003	Serwer i aplikacja przeglądarkowa muszą być wdrażane jako kontenery Docker	Wymagane
INF-004	Baza danych PostgreSQL musi działać jako osobny kontener lub usługa zarządzana	Wymagane
5.2. CI/CD		
ID	Wymaganie	Priorytet
INF-010	System musi posiadać skonfigurowany pipeline CI/CD (GitHub Actions lub odpowiednik)	Wymagane
INF-011	Pipeline musi obejmować: budowanie, uruchamianie testów, budowanie obrazów Docker, wdrażanie na środowisko staging	Wymagane
INF-012	Wdrożenie na produkcję musi wymagać manualnego zatwierdzenia	Wymagane
5.3. Monitoring i logowania		
ID	Wymaganie	Priorytet
INF-020	Serwer musi logować zdarzenia systemowe (połączenia, błędy, zdarzenia scenariuszy) do ustrukturyzowanych logów	Wymagane
INF-021	System musi posiadać monitoring dostępności (health check) z powiadomieniami o awariach (e-mail lub webhook)	Wymagane
INF-022	System powinien udostępniać metryki wydajności (czas odpowiedzi API, liczba aktywnych połączeń, zużycie zasobów)	Wymagane
INF-023	Logi muszą być przechowywane przez minimum 30 dni	Wymagane
5.4. Kopie zapasowe		
ID	Wymaganie	Priorytet
INF-030	Baza danych musi posiadać automatyczne kopie zapasowe wykonywane co najmniej raz dziennie	Wymagane
INF-031	Kopie zapasowe muszą być przechowywane przez minimum 30 dni	Wymagane
INF-032	Musi istnieć udokumentowana procedura przywracania bazy danych z kopii zapasowej	Wymagane
INF-033	Procedura przywracania musi być przetestowana co najmniej raz przed odbiorem końcowym	Wymagane
6. Utrzymanie i wsparcie (24 miesiące)		
6.1. Zakres utrzymania		
ID	Wymaganie	Priorytet
UTR-001	Wykonawca zobowiązany jest do świadczenia usług utrzymaniowych przez 24 miesiące od daty odbioru końcowego	Wymagane
UTR-002	Zakres utrzymania obejmuje: zapewnienie dostępności systemu, poprawki błędów, aktualizacje bezpieczeństwa	Wymagane
UTR-003	Wykonawca zobowiązany jest do aktualizacji zależności (SDK, frameworki) w	Wymagane



	zakresie wersji patch i minor w ramach utrzymania		
UTR-004	Wykonawca zobowiązany jest do dostosowania aplikacji AR do aktualizacji firmware gogli wirtualnej rzeczywistości Meta Quest 3, jeśli aktualizacja wpływa na działanie aplikacji		Wymagane
6.2. Poziomy SLA			
Priorytet	Opis	Czas reakcji	Czas rozwiązania
Krytyczny	System niedostępny lub transmisja niemożliwa	4 godziny robocze	1 dzień roboczy
Wysoki	Funkcjonalność scenariuszy lub telemetrii poważnie ograniczona	8 godzin roboczych	3 dni robocze
Normalny	Drobne błędy interfejsu, problemy niekrytyczne	2 dni robocze	10 dni roboczych
Niski	Uwagi kosmetyczne, propozycje poprawek	5 dni roboczych	Następna planowana aktualizacja
6.3. Dostępność			
ID	Wymaganie		Priorytet
UTR-010	Wsparcie dostępne w godzinach roboczych (poniedziałek–piątek, 8:00–16:00)		Wymagane
UTR-011	Kanał zgłoszeń: e-mail		
UTR-012	Raport z realizacji SLA i wykonanych prac utrzymaniowych na żądanie Zamawiającego		

* Interface w j. angielskim