Codenames Game - Development Plan

1. Overview

This document outlines the structured development plan for the Codenames game. The project will integrate React for the frontend, Firebase for real-time turn management, ASP.NET Core for backend logic, and SQL Server for structured data storage.

2. Development Phases

Phase 1 - Setting Up SQL Server Database

Objective: Establish a structured relational database for user management, game records, and statistics.

Tasks:

- Create SQL Server tables:

  - Users (User management)

  - Games (Game sessions)

  - PlayersInGame (Mapping players to games)

  - Cards (Game cards)

  - Moves (Tracking player actions)

  - ChatMessages (Game chat logs)

- Test database by inserting and querying sample data.

Phase 2 - Developing Backend with ASP.NET Core

Objective: Implement an API to facilitate data exchange between the frontend and the database.

Tasks:

- Set up an ASP.NET Core Web API project.

- Configure Entity Framework Core to connect to SQL Server.

- Develop DBServices for handling data operations.

- Create API Controllers:

  - /api/auth (User authentication and registration)

  - /api/game (Game creation and management)

  - /api/cards (Fetching and updating cards)

  - /api/chat (Handling chat messages)

Phase 3 - Integrating Firebase for Real-Time Game Updates

Objective: Manage live game state, turns, and updates using Firebase Realtime Database.

Tasks:

- Configure Firebase Realtime Database.

- Store game state, current turn, and revealed cards.

- Develop functions to listen for changes:

  - onTurnChange() - Updates frontend UI when the turn changes.

  - onCardReveal() - Syncs card selection across all players.

- Implement API connection to update both Firebase and SQL Server when necessary.

Phase 4 - Developing Frontend with React

Objective: Build an interactive and responsive user interface using React.

Tasks:

- Install React + Tailwind CSS.

- Implement core pages:

- LoginPage (Firebase authentication)

  - LobbyPage (Game list and player selection)

  - GamePage (Game board with real-time interaction)

- Connect React components to Firebase listeners and ASP.NET API calls.

Phase 5 - Enhancements and Optimization

Objective: Improve game experience with advanced features and performance optimization.

Tasks:

- Enhance UI/UX with animations and responsive design.

- Implement game-winning conditions detection.

- Optimize API calls to reduce latency.

- Conduct performance and security testing.

Step-by-Step Execution Strategy

- Each phase is tested before moving to the next.

- Firebase handles real-time updates, while SQL Server stores structured data.

- ASP.NET API ensures smooth backend operations.

- React UI dynamically updates with Firebase changes.

Next Steps

* Should we start with SQL Server database setup or API development first?

* Would you like help with Firebase Realtime Database configuration?

Let me know how you'd like to proceed!