

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Зв'язок на відстані: Гра для пар</title>
  <!-- Tailwind CSS CDN -->
  <script src="https://cdn.tailwindcss.com"></script>
  <!-- Google Fonts: Inter -->
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&
display=swap" rel="stylesheet">
  <style>
    body {
      font-family: 'Inter', sans-serif;
      background-color: #fce4ec; /* Ніжний рожевий фон */
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      margin: 0;
      padding: 1rem; /* Додаємо відступ для мобільних пристроїв
*/
      box-sizing: border-box;
    }
    .game-container {
      background-color: #ffffff; /* Білий фон картки */
      box-shadow: 0 15px 30px rgba(0, 0, 0, 0.15); /* Більш
виразна тінь */
      transition: transform 0.3s ease-in-out;
      max-width: 700px; /* Обмежуємо ширину для кращого вигляду
на великих екранах */
      width: 100%; /* Повна ширина на мобільних */
    }
    .game-container:hover {
      transform: translateY(-8px); /* Більший ефект підняття */
    }
    .question-box {
      min-height: 150px; /* Достатня висота для запитання */
      display: flex;
      justify-content: center;
      align-items: center;
      text-align: center;
      font-size: 1.5rem; /* Більший розмір шрифту для запитання
*/
      font-weight: 600; /* Напівжирний шрифт */
      color: #880e4f; /* Темно-рожевий текст */

```

```

        opacity: 0;
        transition: opacity 0.7s ease-in-out; /* Повільніша
анімація */
        line-height: 1.4; /* Міжрядковий інтервал */
    }
    .question-box.show {
        opacity: 1;
    }
    .button-love-gradient {
        background-image: linear-gradient(to right, #ec4899,
#f472b6); /* Градієнт від рожевого до світло-рожевого */
        transition: all 0.3s ease;
        box-shadow: 0 5px 15px rgba(236, 72, 153, 0.4); /* Тінь
кнопки */
    }
    .button-love-gradient:hover {
        transform: scale(1.05); /* Збільшення при наведенні */
        box-shadow: 0 8px 20px rgba(236, 72, 153, 0.6);
    }
    .answer-box {
        background-color: #ffebee; /* Дуже світлий рожевий для
відповідей */
        border: 1px solid #f8bbd0; /* Тонка рожева рамка */
        min-height: 80px;
        display: flex;
        justify-content: center;
        align-items: center;
        text-align: center;
        font-size: 1.1rem;
        color: #4a148c; /* Темно-фіолетовий для відповідей */
        word-wrap: break-word; /* Перенос слів */
        white-space: pre-wrap; /* Зберігає форматування тексту */
    }
    .loading-spinner {
        border: 4px solid #f3f3f3;
        border-top: 4px solid #ec4899;
        border-radius: 50%;
        width: 30px;
        height: 30px;
        animation: spin 1s linear infinite;
        margin: 0 auto;
    }
    @keyframes spin {
        0% { transform: rotate(0deg); }
        100% { transform: rotate(360deg); }
    }
    /* Custom Modal for Messages */
    #messageModal {

```

```

        background-color: rgba(0, 0, 0, 0.5);
    }
    /* Адаптивність для менших екранів */
    @media (max-width: 768px) {
        .question-box {
            font-size: 1.25rem;
            min-height: 120px;
        }
        .game-container {
            padding: 1.5rem;
        }
        .answer-input {
            font-size: 0.9rem;
        }
    }
</style>
</head>
<body>
    <div class="game-container p-6 md:p-8 rounded-3xl">
        <h1 class="text-4xl font-extrabold text-center text-pink-700
mb-4">Зв'язок на відстані</h1>
        <p class="text-center text-pink-500 mb-8 text-lg">
            Гра для двох гравців. Створіть гру або приєднайтеся до
існуючої за ID.
        </p>

        <!-- Секція для входу/створення гри -->
        <div id="gameIdSection" class="mb-8 p-4 bg-pink-100 rounded-lg
text-center shadow-inner">
            <p class="text-pink-700 font-semibold mb-3">Ваш ID
користувача: <span id="currentUserId" class="font-bold text-pink-900
break-all">Завантаження...</span></p>
            <p class="text-pink-700 font-semibold mb-2">Щоб грати,
потрібен ID гри:</p>
            <div id="gameIdDisplay" class="text-2xl font-bold
text-pink-900 mb-4 hidden break-all"></div>
            <input type="text" id="gameIdInput" placeholder="Введіть
ID гри або залиште порожнім" class="w-full p-3 border border-pink-300
rounded-lg text-center mb-3 focus:outline-none focus:ring-2
focus:ring-pink-400">
            <button id="joinOrCreateGameButton"
class="button-love-gradient text-white font-bold py-3 px-8
rounded-full w-full">
                Приєднатися до гри / Створити нову
            </button>
        </div>

        <!-- Секція гри (прихована до приєднання) -->

```

```

        <div id="gamePlaySection" class="hidden">
            <div id="questionBox" class="question-box p-6 bg-pink-50
rounded-2xl mb-8">
                <!-- Тут з'явиться запитання -->
            </div>

            <!-- Секція для відповіді поточного партнера -->
            <div class="mb-6">
                <label for="myAnswerInput" class="block text-pink-600
text-lg font-semibold mb-2">Ваша відповідь:</label>
                <textarea id="myAnswerInput" class="answer-input
w-full p-3 border border-pink-300 rounded-lg focus:outline-none
focus:ring-2 focus:ring-pink-400 resize-y" rows="3"
placeholder="Напишіть свою відповідь тут..."></textarea>
                <button id="submitMyAnswerButton"
class="button-love-gradient text-white font-bold py-2 px-6
rounded-full mt-3 w-full">
                    Надіслати мою відповідь
                </button>
                <p id="partnerStatus" class="text-center text-pink-600
mt-3 hidden">Очікування відповіді партнера...</p>
            </div>

            <!-- Секція для відображення відповідей (прихована до
обміну) -->
            <div id="revealedAnswersSection" class="hidden">
                <div class="mb-6">
                    <label class="block text-pink-600 text-lg
font-semibold mb-2">Ваша відповідь:</label>
                    <div id="myAnswerDisplay" class="answer-box p-3
rounded-lg"></div>
                </div>
                <div class="mb-6">
                    <label class="block text-pink-600 text-lg
font-semibold mb-2">Відповідь партнера:</label>
                    <div id="partnerAnswerDisplay" class="answer-box
p-3 rounded-lg"></div>
                </div>
            </div>

            <div class="flex justify-center mt-8">
                <button id="exchangeAnswersButton"
class="button-love-gradient text-white font-bold py-3 px-8
rounded-full text-lg shadow-lg focus:outline-none focus:ring-4
focus:ring-pink-300 focus:ring-opacity-75 hidden">
                    Обмінятися відповідями
                </button>
            </div>

```

```

        <div class="flex justify-center mt-8">
            <button id="nextQuestionButton"
class="button-love-gradient text-white font-bold py-4 px-10
rounded-full text-xl focus:outline-none focus:ring-4
focus:ring-pink-300 focus:ring-opacity-75 hidden">
                Наступне запитання
            </button>
        </div>
    </div>
</div>

<!-- Custom Modal for Messages -->
<div id="messageModal" class="fixed inset-0 bg-gray-600
bg-opacity-50 flex items-center justify-center hidden z-50">
    <div class="bg-white p-6 rounded-lg shadow-xl max-w-sm w-full
text-center">
        <h3 id="messageModalTitle" class="text-xl font-semibold
mb-4 text-gray-800"></h3>
        <p id="messageModalContent" class="text-gray-700
mb-6"></p>
        <button id="messageModalCloseButton"
class="button-love-gradient text-white font-bold py-2 px-6
rounded-full">
            Закрити
        </button>
    </div>
</div>

<script type="module">
    // Firebase imports
    import { initializeApp } from
"https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";
    import { getAuth, signInAnonymously, signInWithCustomToken,
onAuthStateChanged } from
"https://www.gstatic.com/firebasejs/11.6.1/firebase-auth.js";
    import { getFirestore, doc, getDoc, setDoc, onSnapshot,
collection } from
"https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore.js";

    // Global variables for Firebase config and app ID
    const appId = typeof __app_id !== 'undefined' ? __app_id :
'default-app-id';
    const firebaseConfig = typeof __firebase_config !==
'undefined' ? JSON.parse(__firebase_config) : {};
    const initialAuthToken = typeof __initial_auth_token !==
'undefined' ? __initial_auth_token : null;

```

```

// Firebase app and services
let app;
let db;
let auth;
let currentUserId = null;
let gameId = null;
let gameDocRef = null;
let currentQuestionIndex = 0;
let unsubscribeGameListener = null;
let unsubscribeAnswerListener = null;

// UI Elements
const gameIdSection =
document.getElementById('gameIdSection');
const gameIdDisplay =
document.getElementById('gameIdDisplay');
const gameIdInput = document.getElementById('gameIdInput');
const joinOrCreateGameButton =
document.getElementById('joinOrCreateGameButton');
const gamePlaySection =
document.getElementById('gamePlaySection');
const questionBox = document.getElementById('questionBox');
const myAnswerInput =
document.getElementById('myAnswerInput');
const submitMyAnswerButton =
document.getElementById('submitMyAnswerButton');
const partnerStatus =
document.getElementById('partnerStatus');
const revealedAnswersSection =
document.getElementById('revealedAnswersSection');
const myAnswerDisplay =
document.getElementById('myAnswerDisplay');
const partnerAnswerDisplay =
document.getElementById('partnerAnswerDisplay');
const exchangeAnswersButton =
document.getElementById('exchangeAnswersButton');
const nextQuestionButton =
document.getElementById('nextQuestionButton');
const currentUserIdSpan =
document.getElementById('currentUserId');

// Custom Modal Elements
const messageModal = document.getElementById('messageModal');
const messageModalTitle =
document.getElementById('messageModalTitle');
const messageModalContent =
document.getElementById('messageModalContent');
const messageModalCloseButton =

```

```

document.getElementById('messageModalCloseButton');

// Questions for the game
const questions = [
    "Яка наша улюблена спільна пам'ять, і чому вона для тебе особлива?",
    "Що ти найбільше цінуєш у наших стосунках на відстані?",
    "Яка одна річ, яку ти хотів би зробити зі мною, коли ми зустрінемося наступного разу?",
    "Як ти уявляєш наше ідеальне побачення на відстані?",
    "Яка пісня нагадує тобі про мене або про наші стосунки?",
    "Що змушує тебе посміхатися, коли ти думаєш про мене?",
    "Який твій улюблений спосіб відчувати зв'язок, коли ми далеко?",
    "Як ти думаєш, яка моя улюблена риса в тобі?",
    "Яке твоє найсмішніше спогад про нас?",
    "Як ти уявляєш наше спільне майбутнє?",
    "Яка одна річ, яку ти дізнався про себе завдяки нашим стосункам на відстані?",
    "Як ти справляєшся з труднощами, пов'язаними з відстанню?",
    "Яка твоя улюблена річ, яку ми робимо разом, навіть якщо це віртуально?",
    "Як ти можеш підтримати мене, коли я почуваюся самотньо?",
    "Яка твоя улюблена якість у мені, яку ти, можливо, не помітив би, якби ми не були на відстані?",
    "Який твій улюблений спосіб провести час, коли ми разом?",
    "Що ти з нетерпінням чекаєш найбільше, коли ми знову побачимося?",
    "Яка твоя улюблена мрія, яку ти хотів би розділити зі мною?",
    "Що ти найбільше цінуєш у нашій комунікації?",
    "Як ти думаєш, що робить наші стосунки такими міцними, незважаючи на відстань?"
];

// --- Utility Functions ---
function showMessageModal(title, content) {
    messageModalTitle.textContent = title;
    messageModalContent.textContent = content;
    messageModal.classList.remove('hidden');
}

messageModalCloseButton.addEventListener('click', () => {
    messageModal.classList.add('hidden');
});

```

```

        function generateGameId() {
            return Math.random().toString(36).substring(2,
8).toUpperCase();
        }

// --- Firebase Initialization and Authentication ---
async function initializeFirebase() {
    try {
        app = initializeApp(firebaseConfig);
        db = getFirestore(app);
        auth = getAuth(app);

        onAuthStateChanged(auth, async (user) => {
            if (user) {
                currentUserId = user.uid;
                currentUserIdSpan.textContent = currentUserId;
                console.log("Authenticated as:",
currentUserId);
            } else {
                // Sign in anonymously if no custom token is
provided or user is not authenticated
                if (initialAuthToken) {
                    await signInWithCustomToken(auth,
initialAuthToken);
                } else {
                    await signInAnonymously(auth);
                }
            }
        });
    } catch (error) {
        console.error("Error initializing Firebase:", error);
        showMessageModal("Помилка Firebase", "Не вдалося
ініціалізувати Firebase. Перевірте консоль для деталей.");
    }
}

// --- Game Logic ---

async function joinOrCreateGame() {
    if (!currentUserId) {
        showMessageModal("Зачекайте", "Будь ласка, зачекайте,
поки користувач буде автентифікований.");
        return;
    }

    let inputId = gameIdInput.value.trim();
    if (inputId) {

```



```

        gameId = inputId;
    } else {
        gameId = generateGameId();
    }

    gameDocRef = doc(db,
`artifacts/${appId}/public/data/longDistanceLoveGames`, gameId);

    try {
        const gameDocSnap = await getDoc(gameDocRef);

        if (!gameDocSnap.exists()) {
            // Create new game
            await setDoc(gameDocRef, {
                currentQuestionIndex: 0,
                partner1Id: currentUserId,
                partner2Id: null,
                lastUpdated: Date.now()
            });
            showMessageModal("Гра створена", `Ваш ID гри:
${gameId}. Поділіться ним з партнером!`);
        } else {
            // Join existing game
            const gameData = gameDocSnap.data();
            if (gameData.partner1Id === currentUserId ||
gameData.partner2Id === currentUserId) {
                // Already part of this game
                showMessageModal("Приєднано", `Ви вже є
частиною гри з ID: ${gameId}.`);
            } else if (!gameData.partner2Id) {
                // Partner 2 joins
                await setDoc(gameDocRef, { partner2Id:
currentUserId, lastUpdated: Date.now() }, { merge: true });
                showMessageModal("Приєднано", `Ви приєдналися
до гри з ID: ${gameId}.`);
            } else {
                // Game is full
                showMessageModal("Гра повна", `Гра з ID:
${gameId} вже має двох гравців.`);
                gameId = null; // Reset gameId
                return;
            }
        }

        gameIdDisplay.textContent = `ID гри: ${gameId}`;
        gameIdDisplay.classList.remove('hidden');
        gameIdSection.classList.add('hidden'); // Hide game ID
    }
}

input

```

```

        gamePlaySection.classList.remove('hidden'); // Show
game play section
        listenToGameChanges(); // Start listening for game
state changes

    } catch (error) {
        console.error("Error joining/creating game:", error);
        showMessageModal("Помилка", "Не вдалося приєднатися
або створити гру. Спробуйте ще раз.");
        gameId = null; // Reset gameId on error
    }
}

function listenToGameChanges() {
    if (unsubscribeGameListener) {
        unsubscribeGameListener(); // Unsubscribe from
previous listener if any
    }
    unsubscribeGameListener = onSnapshot(gameDocRef, (docSnap)
=> {
        if (docSnap.exists()) {
            const gameData = docSnap.data();
            currentQuestionIndex =
gameData.currentQuestionIndex || 0;
            // Ensure both partners are present before
starting game logic
            if (gameData.partner1Id && gameData.partner2Id) {
                displayQuestion();
                listenToAnswerChanges(currentQuestionIndex);
            } else {
                questionBox.textContent = "Очікування другого
партнера...";
                questionBox.classList.add('show');
                // Hide all game play elements until both
partners are connected
                myAnswerInput.classList.add('hidden');
                submitMyAnswerButton.classList.add('hidden');
                partnerStatus.classList.add('hidden');

                revealedAnswersSection.classList.add('hidden');
                exchangeAnswersButton.classList.add('hidden');
                nextQuestionButton.classList.add('hidden');
            }
        } else {
            showMessageModal("Помилка гри", "Гра не знайдена.
Спробуйте створити нову або приєднатися до іншої.");
            resetGameUI();
        }
    }
}

```

```

    });
}

async function displayQuestion() {
    // Reset UI for new question
    questionBox.classList.remove('show');
    myAnswerInput.value = '';
    myAnswerInput.classList.remove('hidden');
    submitMyAnswerButton.classList.remove('hidden');
    submitMyAnswerButton.disabled = false;
    submitMyAnswerButton.textContent = 'Надіслати мою
Відповідь';
    partnerStatus.classList.add('hidden');
    revealedAnswersSection.classList.add('hidden');
    exchangeAnswersButton.classList.add('hidden');
    nextQuestionButton.classList.add('hidden');
    myAnswerDisplay.textContent = '';
    partnerAnswerDisplay.textContent = '';

    // Show question with animation
    setTimeout(() => {
        if (currentQuestionIndex < questions.length) {
            questionBox.textContent =
questions[currentQuestionIndex];
        } else {
            questionBox.textContent = "Ви відповіли на всі
запитання! Чудова робота!";
            nextQuestionButton.classList.add('hidden'); //
Hide next question button at the end
            showMessageModal("Кінець гри", "Ви відповіли на
всі запитання! Дякуємо за гру!");
        }
        questionBox.classList.add('show');
    }, 500);

    // Start listening for answers for the current question
    listenToAnswerChanges(currentQuestionIndex);
}

async function submitMyAnswer() {
    if (!gameId || !currentUserId) {
        showMessageModal("Помилка", "Будь ласка, приєднайтеся
до гри спочатку.");
        return;
    }
    if (myAnswerInput.value.trim() === '') {
        showMessageModal("Увага", "Будь ласка, напишіть свою

```

```

    відповідь.");
        return;
    }

    const answerDocRef = doc(db,
`artifacts/${appId}/public/data/longDistanceLoveGames/${gameId}/answers`, String(currentQuestionIndex));
    const gameData = (await getDoc(gameDocRef)).data();

    let updateData = {};
    if (currentUserId === gameData.partner1Id) {
        updateData.partner1Answer =
myAnswerInput.value.trim();
    } else if (currentUserId === gameData.partner2Id) {
        updateData.partner2Answer =
myAnswerInput.value.trim();
    } else {
        showMessageModal("Помилка", "Ви не є учасником цієї гри.");
        return;
    }

    try {
        await setDoc(answerDocRef, updateData, { merge: true
});

        submitMyAnswerButton.disabled = true;
        submitMyAnswerButton.textContent = 'Відповідь
надіслано!';

        partnerStatus.classList.remove('hidden'); // Show
waiting status
    } catch (error) {
        console.error("Error submitting answer:", error);
        showMessageModal("Помилка", "Не вдалося надіслати
відповідь. Спробуйте ще раз.");
    }
}

function listenToAnswerChanges(qIndex) {
    if (unsubscribeAnswerListener) {
        unsubscribeAnswerListener(); // Unsubscribe from
previous listener
    }
    const answerDocRef = doc(db,
`artifacts/${appId}/public/data/longDistanceLoveGames/${gameId}/answers`, String(qIndex));
    unsubscribeAnswerListener = onSnapshot(answerDocRef, async
(docSnap) => {
        if (docSnap.exists()) {

```

```

        const answers = docSnap.data();
        const gameData = (await
getDoc(gameDocRef)).data();

        const myAnswerField = currentUserId ===
gameData.partner1Id ? 'partner1Answer' : 'partner2Answer';
        const partnerAnswerField = currentUserId ===
gameData.partner1Id ? 'partner2Answer' : 'partner1Answer';

        // Check if both answers are present and not
revealed yet
        if (answers[myAnswerField] &&
answers[partnerAnswerField] && !answers.revealed) {

exchangeAnswersButton.classList.remove('hidden');
        partnerStatus.classList.add('hidden'); // Hide
waiting status
        } else if (answers.revealed) {
            // If answers are already revealed, display
them
            displayRevealedAnswers(answers, gameData);
        }
    } else {
        // No answers yet for this question
exchangeAnswersButton.classList.add('hidden');
partnerStatus.classList.add('hidden');
revealedAnswersSection.classList.add('hidden');
nextQuestionButton.classList.add('hidden');
    }
});
}

async function exchangeAnswers() {
    if (!gameId || !currentUserId) return;

    const answerDocRef = doc(db,
`artifacts/${appId}/public/data/longDistanceLoveGames/${gameId}/answer
s`, String(currentQuestionIndex));
    try {
        await setDoc(answerDocRef, { revealed: true }, {
merge: true });
        // The onSnapshot listener will then trigger
displayRevealedAnswers
    } catch (error) {
        console.error("Error setting revealed status:",
error);
        showMessageModal("Помилка", "Не вдалося обмінятися
відповідями.");
    }
}

```

```

    }
  }

  async function displayRevealedAnswers(answers, gameData) {
    const myAnswerField = currentUserId ===
gameData.partner1Id ? 'partner1Answer' : 'partner2Answer';
    const partnerAnswerField = currentUserId ===
gameData.partner1Id ? 'partner2Answer' : 'partner1Answer';

    myAnswerInput.classList.add('hidden'); // Hide input field
    submitMyAnswerButton.classList.add('hidden'); // Hide
submit button
    exchangeAnswersButton.classList.add('hidden'); // Hide
exchange button

    myAnswerDisplay.textContent = answers[myAnswerField] ||
"Відповідь відсутня";
    partnerAnswerDisplay.textContent =
answers[partnerAnswerField] || "Відповідь відсутня";
    revealedAnswersSection.classList.remove('hidden'); // Show
revealed answers

    nextQuestionButton.classList.remove('hidden'); // Show
next question button
  }

  async function nextQuestion() {
    if (!gameId) return;

    const newQuestionIndex = currentQuestionIndex + 1;
    if (newQuestionIndex < questions.length) {
      try {
        await setDoc(gameDocRef, { currentQuestionIndex:
newQuestionIndex, lastUpdated: Date.now() }, { merge: true });
        // The onSnapshot listener will handle displaying
the new question
      } catch (error) {
        console.error("Error moving to next question:",
error);
        showMessageModal("Помилка", "Не вдалося перейти до
наступного запитання.");
      }
    } else {
      showMessageModal("Кінець гри", "Ви відповіли на всі
запитання! Чудова робота!");
      nextQuestionButton.classList.add('hidden');
    }
  }
}

```

```

function resetGameUI() {
  gameIdSection.classList.remove('hidden');
  gamePlaySection.classList.add('hidden');
  gameIdInput.value = '';
  gameIdDisplay.classList.add('hidden');
  questionBox.textContent = '';
  questionBox.classList.remove('show');
  myAnswerInput.value = '';
  myAnswerInput.classList.remove('hidden');
  submitMyAnswerButton.classList.remove('hidden');
  partnerStatus.classList.add('hidden');
  revealedAnswersSection.classList.add('hidden');
  exchangeAnswersButton.classList.add('hidden');
  nextQuestionButton.classList.add('hidden');
  myAnswerDisplay.textContent = '';
  partnerAnswerDisplay.textContent = '';
  if (unsubscribeGameListener) unsubscribeGameListener();
  if (unsubscribeAnswerListener)
unsubscribeAnswerListener();
  gameId = null;
  currentQuestionIndex = 0;
}

// --- Event Listeners ---
joinOrCreateGameButton.addEventListener('click',
joinOrCreateGame);
submitMyAnswerButton.addEventListener('click',
submitMyAnswer);
exchangeAnswersButton.addEventListener('click',
exchangeAnswers);
nextQuestionButton.addEventListener('click', nextQuestion);

// Initialize Firebase on window load
window.onload = initializeFirebase;
</script>
</body>
</html>

```