```
export async function createFeedback(params: CreateFeedbackParams) {
	const { interviewId, userId, transcript, feedbackId, candidateCode } = params;
	console.log("Interview ID:", interviewId);

	try {
	const formattedTranscript = formatTranscript(transcript);
	const prompt = buildPrompt(formattedTranscript, candidateCode);

	const { object } = await generateObject({
	model: google("gemini-2.0-flash-001", {
		structuredOutputs: false,
	}),
	schema: feedbackSchema,
	prompt,
	system: getSystemPrompt(candidateCode),
	});

	const feedback = {
	interviewId,
	userId,
	totalScore: object.totalScore,
	categoryScores: object.categoryScores,
	strengths: object.strengths,
	areasForImprovement: object.areasForImprovement,
	finalAssessment: object.finalAssessment,
	...(candidateCode && { candidateCode }),
	createdAt: new Date().toISOString(),
	};

	const feedbackRef = feedbackId
	? db.collection("feedback").doc(feedbackId)
	: db.collection("feedback").doc();

	await feedbackRef.set(feedback);

	return { success: true, feedbackId: feedbackRef.id };
	} catch (error) {
	console.error("Error creating feedback:", error);
	return { success: false };
	}
}

// Helper to format transcript
function formatTranscript(transcript: { role: string; content: string }[]): string {
```

```
        return transcript
        .map(({ role, content }) => `- ${role}: ${content}`)
        .join("\n");
}

// Helper to construct prompt
function buildPrompt(formattedTranscript: string, candidateCode?: CandidateCode): string {
        let prompt = `
You are an AI interviewer analyzing a mock interview. Your task is to evaluate the candidate based on structured categories. Be thorough and detailed in your analysis. Don't be lenient with the candidate. If there are mistakes or areas for improvement, point them out.

Transcript:
${formattedTranscript}
        `;

        if (candidateCode) {
        console.log("Candidate Code:", candidateCode);
        prompt += `

${candidateCode.heading}:
Language: ${candidateCode.language}
Code:
\`\`\`${candidateCode.language}
${candidateCode.code}
\`\`\`

Please evaluate the code quality, logic, efficiency, and best practices in addition to the conversation.`;
        }

        prompt += `

Please score the candidate from 0 to 100 in the following areas. Do not add categories other than the ones provided:
- **Communication Skills**: Clarity, articulation, structured responses.
- **Technical Knowledge**: Understanding of key concepts for the role.
- **Problem-Solving**: Ability to analyze problems and propose solutions.
- **Cultural & Role Fit**: Alignment with company values and job role.
- **Confidence & Clarity**: Confidence in responses, engagement, and clarity.
        `;

        return prompt;
}
```

```
// Helper to return the appropriate system instruction
function getSystemPrompt(candidateCode?: CandidateCode): string {
	return candidateCode
	? "You are a professional interviewer analyzing a mock interview that includes both conversation and coding assessment. Evaluate both verbal responses and code quality comprehensively."
	: "You are a professional interviewer analyzing a mock interview. Your task is to evaluate the candidate based on structured categories.";
}
```