**CENTRO UNIVERSITÁRIO DE CIÊNCIAS E TECNOLOGIA DO MARANHÃO**

**– UNIFACEMA**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

AUGUSTO CÉSAR EVANGELISTA DOS SANTOS

CHRISTIAN SAMUEL SAMPAIO

DEBORAH CRISTINA VIEIRA TRINDADE

PHELLIPE DUARTE ARAÚJO

GUILHERME PEREIRA DA SILVA JULIANA LAYARA SANTOS

THUAN CAIQUE LIMA DE SOUSA

**URL do site:** [**https://click-estudos-64a85.web.app**](https://click-estudos-64a85.web.app)

**Github:**

**https://github.com/Sampaioooo/PIE-PROJETO.git**

**PROJETO INTEGRADOR EXTENSIONISTA – DESENVOLVIMENTO BACKEND**

**ETAPA 3: ENTREGA DO PROJETO**

Trabalho da Disciplina Projeto Integrador Extensionista – Desenvolvimento Back-End, o Professor: Marcos Gomes, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas - UNIFACEMA.

**CAXIAS -MA**

**(2024)**

**Documentação do Projeto: Software Educacional**

Visão Geral do Projeto

O projeto é um software educacional voltado para crianças, que combina vídeos, cursos, quizzes e jogos didáticos. O objetivo é criar um ambiente interativo e divertido para o aprendizado, semelhante a plataformas como Duolingo, mas com um foco mais amplo em diversas disciplinas. O sistema é dividido em front-end e back-end, cada um desempenhando um papel crucial na funcionalidade do software.

Back-End

Tecnologias Utilizadas

Firebase: O back-end do projeto é construído sobre a plataforma Firebase, que fornece uma variedade de serviços, incluindo autenticação de usuários, banco de dados em tempo real e armazenamento de arquivos. Isso permite um gerenciamento eficiente de dados e uma experiência de usuário fluida.

JavaScript: A lógica do back-end é implementada em JavaScript, utilizando as bibliotecas do Firebase para interagir com os serviços.

Funcionalidades

Autenticação de Usuários:

O sistema permite que os usuários se registrem e façam login. As credenciais são armazenadas de forma segura no Firebase Authentication.

O código de autenticação verifica se o e-mail e a senha fornecidos correspondem aos dados armazenados, permitindo acesso ao sistema.

Gerenciamento de Cursos e Vídeos:

Os cursos e vídeos são organizados em um banco de dados no Firebase. Cada disciplina possui uma lista de vídeos e perguntas relacionadas.

O sistema permite que os usuários acessem conteúdos específicos de acordo com a disciplina escolhida.

Quizzes e Jogos Didáticos:

Após assistir aos vídeos, os usuários podem participar de quizzes que testam seu conhecimento sobre o conteúdo aprendido.

O back-end gerencia as perguntas do quiz, armazenando as respostas corretas e verificando as respostas dos usuários.

Armazenamento de Dados:

O Firebase Firestore é utilizado para armazenar informações sobre usuários, cursos, vídeos e resultados de quizzes.

Isso permite que o sistema mantenha um histórico de progresso dos usuários e ofereça uma experiência personalizada.

Estrutura de Dados

Usuários: Armazenados com e-mail e senha.

Cursos: Cada curso contém um título e uma lista de vídeos.

Vídeos: Cada vídeo possui um título e um link para o conteúdo no YouTube.

Quizzes: Cada quiz contém perguntas, opções de resposta e a resposta correta.

Front-End

Tecnologias Utilizadas

HTML/CSS: A estrutura e o estilo das páginas são construídos usando HTML e CSS, garantindo uma interface amigável e acessível para crianças.

JavaScript: Utilizado para a lógica de interação do usuário, como exibir vídeos, gerenciar quizzes e validar respostas.

Funcionalidades

Interface de Usuário:

A interface é projetada para ser intuitiva e atraente, com ícones e cores que atraem o público infantil.

Os usuários podem facilmente navegar entre as disciplinas e acessar vídeos e quizzes.

Interatividade:

O front-end permite a interação em tempo real com os usuários, exibindo mensagens de sucesso ou erro com base nas respostas fornecidas nos quizzes.

Jogos didáticos são acessíveis após a conclusão dos quizzes, incentivando o aprendizado através de atividades lúdicas.

Responsividade:

O design é responsivo, permitindo que o software funcione bem em dispositivos móveis e desktops, garantindo acessibilidade para todos os usuários.

**1. Autenticação de Usuários**

**a. Estrutura de Login**

O sistema de autenticação é essencial para garantir que apenas usuários registrados possam acessar o conteúdo. A lógica de login verifica as credenciais fornecidas pelo usuário.

javascript

Copiar código

// Lista de usuários (exemplo)

const usuarios = [

{ email: 'responsavel@example.com', senha: '1234' }

];

// Função de login

function login() {

const email = document.getElementById("email").value;

const password = document.getElementById("password").value;

const errorMessage = document.getElementById("error-message");

// Simula o login verificando e-mail e senha

const usuario = usuarios.find(user => user.email === email && user.senha === password);

if (usuario) {

// Login bem-sucedido, redireciona para a tela principal

localStorage.setItem("usuario", email); // Armazena o e-mail do usuário

window.location.href = "home.html"; // Redireciona para a página inicial

} else {

// Exibe mensagem de erro

errorMessage.textContent = "E-mail ou senha inválidos!";

}

}

Essa função realiza a verificação das credenciais, e em caso de sucesso, armazena o e-mail do usuário no localStorage e redireciona para a página principal. Caso contrário, exibe uma mensagem de erro.

**2. Gerenciamento de Cursos e Vídeos**

**a. Estrutura de Dados**

Os cursos e vídeos são organizados em um objeto JavaScript, o que permite fácil acesso e manipulação. Cada disciplina contém uma lista de vídeos, que são carregados dinamicamente.

// Lista de vídeos por disciplina

const cursos = {

"Língua Portuguesa": [

{ titulo: "Playlist de Língua Portuguesa - 6º ano do Ensino Fundamental", url: "https://www.youtube.com/embed/gdrzFc9rTlY" },

{ titulo: "Substantivo – Língua Portuguesa – 6º ano", url: "https://www.youtube.com/embed/cxqPHPPq1Jc" },

{ titulo: "As Narrativas Ficcionais - Parte 1", url: "https://www.youtube.com/embed/fX1p7FJB5m4" },

{ titulo: "Figuras de Linguagem – Aula 12", url: "https://www.youtube.com/embed/gdrzFc9rTlY" }

],

"História": [

{ titulo: "História do Brasil - A Colonização", url: "https://www.youtube.com/embed/Uq7opea\_7J0" },

{ titulo: "A Independência do Brasil", url: "https://www.youtube.com/embed/IIReUDezw-E" },

{ titulo: "Primeira Guerra Mundial", url: "https://www.youtube.com/embed/5MMJ7qCP55Q" },

{ titulo: "Revolução Francesa", url: "https://www.youtube.com/embed/4CthQxp3h4c" }

],

// ... outras disciplinas

};

Esse formato permite fácil adição de cursos e vídeos, e a estrutura pode ser facilmente expandida conforme o número de disciplinas cresce.

**3. Carregamento de Vídeos**

**a. Função para Carregar Vídeos**

Quando o usuário seleciona uma disciplina, os vídeos correspondentes são carregados e exibidos. Abaixo está o código para exibir e navegar entre os vídeos.

// Função para carregar os vídeos

function carregarVideos(disciplina) {

const videosContainer = document.querySelector(".videos");

const videos = cursos[disciplina] || [];

// Variáveis para controlar o índice do vídeo atual

let indiceAtual = 0;

// Exibe o vídeo atual

function exibirVideo(indice) {

const video = videos[indice];

videosContainer.innerHTML = `

<h3>${video.titulo}</h3>

<iframe width="100%" height="500" src="${video.url}?enablejsapi=1" frameborder="0" allowfullscreen></iframe>

`;

// Atualiza os botões após exibir o vídeo

const botoesContainer = document.createElement("div");

botoesContainer.innerHTML = `

<button id="btnAnterior">Anterior</button>

<button id="btnProximo">Próximo</button>

`;

videosContainer.appendChild(botoesContainer);

// Adiciona eventos de navegação

document.getElementById("btnProximo").addEventListener("click", proximoVideo);

document.getElementById("btnAnterior").addEventListener("click", videoAnterior);

}

// Função para navegar para o próximo vídeo

function proximoVideo() {

indiceAtual++;

if (indiceAtual >= videos.length) {

indiceAtual = videos.length - 1; // Não permite ir além do último vídeo

alert("Você alcançou o último vídeo.");

}

exibirVideo(indiceAtual);

}

// Função para navegar para o vídeo anterior

function videoAnterior() {

indiceAtual--;

if (indiceAtual < 0) {

indiceAtual = 0; // Não permite ir além do primeiro vídeo

alert("Você está no primeiro vídeo.");

}

exibirVideo(indiceAtual);

}

// Exibe o primeiro vídeo ao carregar a disciplina

exibirVideo(indiceAtual);

}

Essa função permite ao usuário navegar pelos vídeos de uma disciplina, com botões de "Próximo" e "Anterior" para controlar a exibição.

**4. Quizzes**

**a. Estrutura de Quiz**

Os quizzes ajudam a testar o conhecimento do aluno sobre o conteúdo dos vídeos assistidos. Eles podem ser apresentados após cada vídeo ou ao final de um módulo.

javascript

Copiar código

// Exemplo de um quiz simples com perguntas e respostas

const quizzes = {

"Língua Portuguesa": [

{

pergunta: "Qual é o significado da palavra 'substantivo'?",

alternativas: ["Nome de uma pessoa", "Nome de um objeto", "Nome de um lugar", "Todas as alternativas estão corretas"],

respostaCorreta: 3

},

{

pergunta: "O que é uma figura de linguagem?",

alternativas: ["Uma metáfora", "Uma analogia", "Uma ironia", "Todas as alternativas estão corretas"],

respostaCorreta: 3

}

],

"História": [

{

pergunta: "Quem foi o responsável pela independência do Brasil?",

alternativas: ["D. Pedro I", "D. João VI", "Napoleão Bonaparte", "Dom Pedro II"],

respostaCorreta: 0

},

{

pergunta: "Em que ano aconteceu a Revolução Francesa?",

alternativas: ["1776", "1789", "1804", "1815"],

respostaCorreta: 1

}

],

// ... outros quizzes

};

// Função para exibir as perguntas do quiz

function iniciarQuiz(disciplina) {

const quizContainer = document.querySelector(".quiz");

const quiz = quizzes[disciplina];

let indicePergunta = 0;

let pontuacao = 0;

function exibirPergunta(indice) {

const pergunta = quiz[indice];

quizContainer.innerHTML = `

<h3>${pergunta.pergunta}</h3>

${pergunta.alternativas.map((alt, i) => `

<button class="alternativa" onclick="verificarResposta(${i}, ${indice})">${alt}</button>

`).join('')}

`;

}

function verificarResposta(respostaEscolhida, indice) {

const pergunta = quiz[indice];

if (respostaEscolhida === pergunta.respostaCorreta) {

pontuacao++;

}

indicePergunta++;

if (indicePergunta < quiz.length) {

exibirPergunta(indicePergunta);

} else {

quizContainer.innerHTML = `

<h3>Fim do quiz!</h3>

<p>Sua pontuação foi: ${pontuacao} de ${quiz.length}</p>

`;

}

}

// Inicia o quiz com a primeira pergunta

exibirPergunta(indicePergunta);

}

Aqui, cada quiz contém perguntas, alternativas e uma resposta correta. Quando o aluno seleciona uma alternativa, a função verificarResposta verifica se está correta e move para a próxima pergunta. Ao final, a pontuação do aluno é exibida.

**5. Jogos Educacionais**

**a. Jogo de Perguntas e Respostas**

Os jogos ajudam a envolver os alunos de uma forma mais interativa e divertida. Um exemplo de jogo seria um simples jogo de perguntas e respostas, onde o aluno ganha pontos por respostas corretas.

javascript

Copiar código

// Função para iniciar o jogo de perguntas

function iniciarJogo() {

const jogoContainer = document.querySelector(".jogo");

const perguntas = [

{ pergunta: "Qual é o maior planeta do sistema solar?", resposta: "Júpiter" },

{ pergunta: "Quem escreveu a obra 'Dom Casmurro'?", resposta: "Machado de Assis" },

{ pergunta: "Quantos continentes existem no mundo?", resposta: "7" }

];

let pontos = 0;

let perguntaAtual = 0;

function mostrarPergunta(indice) {

const pergunta = perguntas[indice];

jogoContainer.innerHTML = `

<h3>${pergunta.pergunta}</h3>

<input type="text" id="resposta" placeholder="Digite sua resposta" />

<button onclick="verificarResposta(${indice})">Responder</button>

`;

}

function verificarResposta(indice) {

const resposta = document.getElementById("resposta").value.trim();

const pergunta = perguntas[indice];

if (resposta.toLowerCase() === pergunta.resposta.toLowerCase()) {

pontos++;

alert("Resposta correta!");

} else {

alert(`Resposta errada! A resposta correta é: ${pergunta.resposta}`);

}

perguntaAtual++;

if (perguntaAtual < perguntas.length) {

mostrarPergunta(perguntaAtual);

} else {

jogoContainer.innerHTML = `

<h3>Fim do Jogo!</h3>

<p>Você acertou ${pontos} de ${perguntas.length} perguntas.</p>

`;

}

}

mostrarPergunta(perguntaAtual);

}

Esse jogo de perguntas e respostas permite ao aluno tentar acertar as questões digitando as respostas. Ao final, o número de acertos é mostrado

1. **Testes Unitários**: O objetivo dos testes unitários é verificar o comportamento de pequenas unidades do código (como funções ou métodos). No seu caso, a função verificarResposta() foi testada individualmente com diferentes entradas, o que ajuda a identificar erros lógicos ou problemas de implementação antes que o sistema seja integrado. Isso permite corrigir problemas de forma isolada, sem interferir no restante do sistema.
2. **Testes de Integração**: Testes de integração são cruciais para garantir que diferentes módulos do sistema funcionem juntos de maneira harmoniosa. Por exemplo, ao testar a integração entre a autenticação do usuário e a recuperação de dados do Firestore, você garante que, uma vez autenticado, o sistema consegue acessar e manipular os dados corretamente. Esses testes ajudam a identificar problemas que surgem quando diferentes componentes interagem, que podem não ser detectados nos testes unitários.
3. **Testes de Usabilidade**: Esses testes focam na experiência do usuário. Eles são realizados para avaliar como os usuários interagem com o sistema e se a interface é intuitiva e fácil de usar. Testes de usabilidade podem ser feitos com usuários reais para observar como eles navegam no sistema, quais funcionalidades podem ser confusas e onde melhorias podem ser feitas. O feedback coletado é fundamental para melhorar a interface e a experiência geral do usuário.

Conclusão

Este projeto educacional é uma ferramenta valiosa para o aprendizado infantil, combinando vídeos, quizzes e jogos em uma plataforma interativa. O uso do Firebase para o back-end permite uma gestão eficiente de dados, enquanto o front-end é projetado para ser amigável e acessível. O objetivo final é proporcionar uma experiência de aprendizado divertida e envolvente para crianças, ajudando-as a adquirir conhecimentos de forma lúdica e eficaz.