



nextPlay

22/11/2024

Lucas Athanasio
Enzo Guerra
Laura Alves

Visão Geral

O projeto consiste em uma rede social voltada para gamers, com foco em interação e compartilhamento de conteúdo. A plataforma permite que os usuários criem postagens com textos, imagens, vídeos ou arquivos, comentem nas postagens de outros usuários, curtam ou descurtam conteúdo e sigam outros jogadores. Além disso, cada usuário possui um perfil com informações básicas, contagem de seguidores e histórico de postagens. O sistema foi desenvolvido com PHP no backend, MySQL para o banco de dados e HTML/CSS/Bootstrap no frontend.

Objetivos

I. Objetivo Geral

Desenvolver uma rede social gamer completa, que permita aos usuários compartilhar conteúdo, interagir com outros jogadores e gerenciar suas próprias postagens e comentários de forma segura.

II. Objetivos Específicos

- Implementar cadastro, login e autenticação segura de usuários.
- Criar feed de postagens com suporte a textos, imagens e vídeos.
- Permitir a interação entre usuários através de comentários, curtidas, dislikes e chats privados.
- Desenvolver perfis de usuário com histórico de postagens e contagem de seguidores.
- Garantir que apenas autores possam editar ou excluir seus posts e comentários.

Requisitos

I. Requisitos Funcionais

- Cadastro de usuário com nome, e-mail, nome de usuário e senha.
- Login de usuário com verificação de credenciais.
- Edição de perfil, incluindo alteração de foto de perfil e informações pessoais.
- Criação de postagens (texto, imagens, links, etc.).
- Edição e exclusão de postagens pelo autor.

- Interações nas postagens, como curtir, descurtir e comentar.
- Edição e exclusão de comentários pelo autor.
- Chat privado entre usuários cadastrados.
- Pesquisa de usuários para encontrar perfis.
- Exibição de feed mostrando as postagens de todos os usuários.

II. Requisitos Não Funcionais

- O sistema foi desenvolvido em PHP, com MySQL para armazenamento de dados.
- O sistema utiliza HTML, CSS e JavaScript para a interface do usuário.
- O sistema armazena senhas de forma criptografada.
- O sistema segue boas práticas de usabilidade (menu intuitivo, feedback de ações, etc.).
- O banco de dados garante integridade referencial (relacionamentos corretos entre tabelas).

Banco de Dados

O banco de dados foi estruturado para suportar as funcionalidades da rede social. Abaixo, as tabelas:

users: tabela que armazena informações dos usuários da rede social.

Campos principais:

- **id**: identificador único do usuário.
- **username**: nome de usuário (único).
- **email**: endereço de e-mail (único).
- **senha**: senha do usuário, armazenada de forma segura.
- **profile_pic**: caminho para a foto de perfil.
- **data_cadastro**: data em que o usuário se cadastrou.
- **tipo**: define se o usuário é 'comum' ou 'administrador'.

Finalidade: controlar o cadastro, autenticação e perfil dos usuários.

jogos: tabela que contém os jogos disponíveis para posts de partidas.

Campos principais:

- **id**: identificador único do jogo.
- **nome**: nome do jogo.

Finalidade: permite criar postagens relacionadas a partidas de jogos específicos.

posts: tabela genérica de postagens, usada para diferenciar tipos de posts.

Campos principais:

- **id**: identificador único da postagem.
- **user_id**: referência ao autor do post (**users.id**).
- **tipo**: define o tipo do post: **normal** ou **jogo**.
- **created_at**: data e hora de criação do post.

Finalidade: centraliza todas as postagens, permitindo diferenciar postagens normais de posts de jogos.

postnormal: tabela específica para posts do tipo normal.

Campos principais:

- **id**: identificador único.
- **post_id**: referência ao post genérico (**posts.id**).
- **titulo**: título do post.
- **conteudo**: texto do post.
- **image_path**: caminho de uma imagem associada.
- **file_path**: caminho de um arquivo associado.
- **youtube_link**: link de vídeo do YouTube.
- **created_at**: data e hora de criação.

Finalidade: armazena o conteúdo detalhado de posts normais.

postjogo: tabela específica para posts relacionados a partidas de jogos.

Campos principais:

- **id**: identificador único.
- **post_id**: referência ao post genérico (**posts.id**).

- `jogo_id`: referência ao jogo (`jogos.id`).
- `descricao`: descrição da partida.
- `data_partida`: data da partida.
- `horario`: horário da partida.

Finalidade: detalhar postagens que convidam para partidas de jogos.

`participantepartida`: tabela que relaciona usuários participantes de uma partida.

Campos principais:

- `id_partida`: referência à partida (`postjogo.id`).
- `id_usuario`: referência ao usuário (`users.id`).

Finalidade: gerenciar quem vai participar de cada partida.

`AdministradorExcluiUser`: registra ações de administradores que excluíram usuários.

Campos principais:

- `id_admin`: administrador responsável.
- `user_id`: usuário excluído.

Finalidade: manter histórico de exclusão de usuários.

`AdministradorExcluiPostagem`: registra ações de administradores que excluíram postagens.

Campos principais:

- `id_admin`: administrador responsável.
- `post_id`: post excluído (`postnormal.id`).

Finalidade: manter histórico de exclusão de postagens.

`links`: armazena links associados a posts normais.

Campos principais:

- `id`: identificador único.
- `post_id`: referência ao post (`postnormal.id`).
- `user_id`: usuário que adicionou o link (`users.id`).

Finalidade: permitir que usuários adicionem links extras aos posts.

`likes`: registra curtidas feitas pelos usuários em posts.

Campos principais:

- `id`: identificador único.
- `user_id`: usuário que curtiu.
- `post_id`: post curtido.
- `created_at`: data e hora da curtida.

Finalidade: contabilizar reações positivas.

`dislikes`: registra dislikes feitos pelos usuários em posts.

Campos principais:

- `id`: identificador único.
- `user_id`: usuário que descurtiu.
- `post_id`: post descurtido.
- `created_at`: data e hora do dislike.

Finalidade: contabilizar reações negativas.

`comments`: armazena comentários dos usuários em posts.

Campos principais:

- `id`: identificador único.
- `user_id`: autor do comentário (`users.id`).
- `post_id`: post comentado (`posts.id`).
- `content`: texto do comentário.

- `created_at`: data e hora do comentário.

Finalidade: gerenciar interações entre usuários via comentários.

`planos`: armazena informações de planos dos usuários (básico, pro, ultimate).

Campos principais:

- `id`: identificador único.
- `tipo_plano`: tipo do plano.
- `user_id`: usuário associado (`users.id`).

Finalidade: controlar planos de assinatura e privilégios diferenciados.

`messages`: armazena mensagens privadas entre usuários.

Campos principais:

- `id`: identificador único.
- `remetente_id`: usuário que envia a mensagem.
- `destinatario_id`: usuário que recebe a mensagem.
- `mensagem`: conteúdo da mensagem.
- `timestamp`: data e hora do envio.

Finalidade: permitir comunicação privada entre usuários.

`seguidores`: tabela que registra seguidores e seguindo.

Campos principais:

- `id_seguidor`: usuário que segue.
- `id_seguindo`: usuário seguido.
- `created_at`: data em que passou a seguir.

Finalidade: gerenciar relações de seguidores e seguidos entre os usuários.

Funcionalidades do Sistema

- Feed de Postagens: exibe todas as postagens dos usuários cadastrados, permitindo visualização de textos, imagens, vídeos e arquivos.
- Interação com Conteúdo: usuários podem comentar, editar e excluir seus próprios comentários, além de curtir ou descurtir postagens de qualquer usuário.
- Perfis de Usuário: cada perfil mostra informações básicas, avatar, quantidade de seguidores e seguindo, e histórico de postagens. Usuários podem seguir ou parar de seguir outros usuários.
- Gerenciamento de Conteúdo: apenas o autor da postagem ou comentário pode editá-los ou removê-los, garantindo segurança e controle sobre o próprio conteúdo.
- Chat: usuários podem conversar entre si indo ao "Bate papo" e procurando na listagem de conversas ou pesquisando por nome, ou ainda acessando a aba de "Convites" e clicando no botão "Conversar" para enviar mensagens a quem criou o convite.