Chave digital

Plataforma de alfabetização digital

Rafael Kendy - Hugo Massaro - Rafael Zaupa

OBJETIVOS

O objetivo principal do projeto é reduzir a exclusão digital por meio de uma plataforma que oferece conteúdos gratuitos e acessíveis para ensinar o uso básico de computadores e da internet. A plataforma será um projeto sem fins lucrativos e de código aberto, permitindo a colaboração da comunidade.

Esses conteúdos serão apresentados em formato de textos, imagens, vídeos e exercícios, de maneira a possibilitar o usuário a aprender de maneira independente e gradual.

Alfabetização digital

- **A** Ensino gratuito
- **B** Código aberto
- **C** Redução da exclusão digital
- **D** Implementar Autorização

Questões

- Usado na página de comunidade para o usuário postar dúvidas no site.
- **Nome do usuário (name: str);**
- **Email (email: str);**
- **Título da dúvida (title: str);**
- **A dúvida em si (question: str);**
- **Uma imagem (image_url: Optional[str] = None).**

Usuário

- Usada para gerenciar o acesso, autenticação e perfil dos visitantes da plataforma.
- ID(id: int): chave gerada automaticamente pelo banco;
- Nome(name: str): nome do usuário;
- Email(email: str): identificador p/ login;
- Senha(hashed_password: str): armazena apenas o hash p/ segurança;
- Descrição(description: str): campo opcional p/ biografia.

Cursos

- 1:N em cascata: cursos, seções, lições, passos; menos progresso. São tudo da tela de tópicos até os conteúdos.
- ID (id: int): Identificador único do curso;
- Título (title: str): Nome do curso;
- Descrição (description: str);
- Ícone (image: str): Nome do caminho ou URL;
- **Seções (sections): cursos 1:n seções, seguinte tabela.**

COMUNIDADE

/comunidade - Rafael Kendy

Método PUT

- Usado para atualizações, após o usuário escrever a dúvida e postar ela com o botão (POST), ela aparece na página pelo método GET.
- A edição abre um modal, pedindo confirmação de email, caso o email esteja correto os dados são carregados para o formulário.
- Método busca a questão pelo seu ID, compara os emails e então atualiza a dúvida, ou lança a mensagem de email incorreto.

```python
#edita a questao, segue a msm logica do email do delete
@app.put("/comunidade/{question_id}")
async def update_question(
    question_id: int,
    email: str = Body(..., embed=True),
    name: str | None = Body(None),
    title: str | None = Body(None),
    question: str | None = Body(None),
    session = Depends(get_session),
):
    q = session.get(Question, question_id) #pega direto pelo id
    if not q:
        raise HTTPException(status_code=404, detail="Questão não encontrada")
    if q.email != email: #email errado
        raise HTTPException(status_code=403, detail="Email incorreto. Você não pode editar esta dúvida.")
    #atualiza os dados, ainda to ignorando a imagem
    if name is not None:
        q.name = name
    if title is not None:
        q.title = title
    if question is not None:
        q.question = question

    session.add(q)
    session.commit()
    session.refresh(q)

    return {"message": "Questão atualizada com sucesso", "question": q}


#=============================================================================================================
```

PERFIL

/users/me - Hugo Massaro

Método PUT

- Utiliza-se a dependência Session do FastAPI para abrir uma conexão segura com o Banco de Dados.
- O sistema usa o token JWT para buscar o usuário no banco (Select(User)), se o usuário não existir no banco a conexão é encerrada
- O backend recebe os novos dados atualiza apenas os campos necessários no objeto python e usa session.add() para preparar a mudança
- O comando session.commit() confirma a gravação da mudança no arquivo do banco e o session.refresh() garante que a API retorne os dados mais recentes salvos.

```python
#endpoint perfil UPDATE
@app.put("/users/me", response_model=UserPublic)
async def update_user_me( user_update: UserUpdate,current_user:
                                Annotated[User, Depends(get_current_active_user)], session: Session = Depends(get_session)):
    # pega os dados a serem atualizados do corpo da requisição,fastAPI valida se o JSON enviado bate com o modelo UserU
    #usa o get_current_active_user para pegar o token do cabeçalho e validar ele, alem de dar o objeto user

    #o current_user ja veio do banco pelo get current_active_user
    # o sqlmodel sabe q esse objeto esta conectado a uma linha especifica da tabela

    #atualiza os campos se o frontend enviou um name q nao eh nulo
    if user_update.name is not None:
        current_user.name = user_update.name#atualiza o name

    #atualiza os campos se o frontend enviou uma descrição q nao eh nula
    if user_update.description is not None:
        current_user.description = user_update.description#atualiza a descrição

    #como current_user ja tem um id, o sqlmodel entende que é uma atualização, n um novo registro
    session.add(current_user) #marca para atualização
    session.commit() #salva no bd
    session.refresh(current_user) #pega os dados atualizados, garantindo q ta corretamente atualizado

    #retorna o usuário com os dados atualizados, o response_model=UserPublic garante que a senha hash não vaze
    return user_to_public(current_user)
#endpoint perfil PUT
```

PASSOS

/cursos/# - Rafael Zaupa

Método POST

- Método para utilização da ferramenta de genIA do Google - o Gemini.
- Enquanto não há um especialista em educação no nosso time, utilizamos a ferramenta para dar uma prévia do que poderia estar escrito.
- Só pode ser incluso via Swagger.
- O prompt é breve e tenta ser o mais claro possível.
- São tratados problemas que a IA tende a insistir em causar, como inclusão de tópicos onde não era desejado.
- Por enquanto, deleta os passos antigos em favor de novos.

```
n in lessons:
(f"Processando lição {lesson.id}: {lesson.title}...")

ove passos antigos
ing_steps = session.exec(select(Step).where(Step.lesson_id == lesson.id)).all()
tep in existing_steps:
ession.delete(step)
on.commit()


 prompt da genai
rompt = (
   f"Você é um professor de informática para idosos, iniciantes e pessoas com dificuldade em computador
   f"Crie um tutorial de 3 passos curtos sobre: '{lesson.title}'. "
   f"Sem introdução, sem conclusão. Apenas os 3 passos. 4 passos se achar estritamente necessário."
   f"Não use números (1., 2.) no início das linhas."


 chama o modelo de IA pela API externa
 se der ruim, vai pro except
esponse = model.generate_content(prompt)
exto_gerado = response.text

 processamento do texto que foi gerado
inhas = texto_gerado.strip().split('\n')
assos_limpos = [linha for linha in linhas if linha.strip()]

or texto_passo in passos_limpos:
   texto_limpo = texto_passo.replace('*', '').replace('-', '').strip()
```