INSTITUTO UNIVERSITÁRIO DE LISBOA

Todas as Categorias | Filtrar

Pesquisar subreddits | Pesquisar | Pesquisar users | Pesquisar Users

/r/Interfaces Web para a Gestão de Dados

Criador: jorge_louca 

Categoria: 🖥️ IT

Programação Web com a framework do Django

/r/Serit.Inc

Criador: grupo_8

Categoria: 🖥️ IT

Um novo site competitivo no mercado acabou de ser lançado, com o suporte da Generative AI do Chat GPT e muito mais!

Trabalho realizado por:

- Diogo Alexandre Alonso de Freitas, 104841, CDC1

- João Francisco Marques Gonçalves da Silva Botas, 104782, CDC1

- Marco Delgado Esperança, 110451, CDC1

Comentar

20 | 0

Autor: grupo_8

Publicado em: 28/10/2023 00:31:28

Editar Post

Introdução e Descrição do projeto

O website desenvolvido para este projeto foi inspirado pelo site Reddit. A escolha do nome foi sugerida através da conjugação entre a palavra SER e "IT", que representa o meio das tecnologias da informação, cruzado às nossas vidas. Para além disso, sempre tivemos em mente de dar o nosso máximo e de realizar o "site mais sério da história", o que contribuiu para uma perfeita ligação entre estas duas ideias. A escolha da extensão ".Inc" está relacionada à ideia de "incorporação", uma estrutura empresarial que leva para investimentos mais seguros e estáveis. Assim, surgiu o **Serit.Inc**, um nome que reflete a seriedade, a tecnologia e a solidez dos princípios empresariais do projeto.

Este *website* permite aos seus utilizadores partilhar tudo aquilo que querem, desde dúvidas sobre algum tema, até *lifestyles* saudáveis ou opiniões sobre filmes. Para isso, basta criar um ***subreddit*** para começar uma nova comunidade, onde diferentes utilizadores poderão interagir entre si de várias maneiras diferentes, através de *posts*, comentários e até um chat geral dentro dessa mesma comunidade, em que todos os utilizadores poderão escrever aquilo que querem e pensam em tempo real!! Em baixo, podemos visualizar um mapa simples do que foi referido acima:

```
Subreddit: /r/programming (Categoria IT) ⇒ Chat geral

      Post: "Alguém me ajude, não consigo usar as imagens do static!"

            Comentário1: "No settings.py tens STATIC_URL = '/static/'?"

            Comentário2: "Também tinha esse problema e isso resolveu!"
```

É importante referir que os utilizadores não estão limitados à interação por texto, existindo a possibilidade de compartilhar e publicar imagens, contribuindo para uma maior aproximação entre estes!

Tipos de utilizadores e privilégios

Utilizadores normais: Os utilizadores normais podem criar *subreddits*, *posts* e comentários; visualizar todos os (*subreddits*) que já tenham sido publicados; dar *like* e *dislike* nos *posts* e comentários que quiserem; apagar os **seus próprios** *subreddits*, *posts* e comentários; personalizar **o seu próprio perfil**, colocando: qualquer imagem pretendida, os seus interesses, etc.; visualizar o perfil de outros utilizadores, onde podem segui-lo, deixar de o seguir ou mandar uma mensagem; filtrar/pesquisar os *subreddits*, seja pelos interesses, ou através de uma barra de pesquisa; conversar pelo *chat*, quer seja o geral do *subreddit*, ou um privado com outro utilizador; comunicar com o Chat GPT que está embutido no *website*.

Administradores: Os administradores (*superusers*) dispõem de todas as permissões de um utilizador normal, mas podem eliminar os subreddits, posts e comentários de qualquer utilizador.

Descrição dos diferentes modelos de dados

O modelo de dados utilizado para suportar a base de dados SQLite do *website* contém as seguintes tabelas:

- **InterestCategory** – esta classe representa categorias de interesses que podem estar associadas a um utilizador ou *subreddit*;
- **RedditUser** – armazena os utilizadores registados, sendo que para além dos tradicionais campos de utilizador, possui o país de origem e o género, de forma que uma equipa de cientistas de dados possam fazer estatísticas *à posteriori;*
- **Subreddit** – representa os *subreddits*, que são comunidades no Serit.Inc;
- **Post** – armazena os *posts* dos utilizadores;
- **Comment –** armazena os comentários dos utilizadores;
- **Message** – armazena cada uma das mensagens trocadas entre os utilizadores num chat ou *subreddit*;
- **Chat** – representa os chats que armazenam as mensagens trocadas entre utilizadores numa conversa privada ou num dado *subreddit*.

Pontos fortes e inovadores

Ao longo dos últimos anos a inteligência artificial tem-se afirmado no nosso quotidiano e na indústria tecnológica no geral. Foi através desta demanda que considerámos uma mais valia ter o Chat GPT implementado no nosso site através de uma *API*, de forma que os utilizadores possam trocar impressões entre eles com recurso a esta funcionalidade.

Como o Serit.Inc procura servir o mundo todo, o Chat GPT permite oferecer uma assistência rápida a 24h / 7 dias por semana, sendo muito valiosa para os membros da comunidade que necessitem de assistência a qualquer momento. Além disso, disponibiliza uma camada adicional de interatividade ao *website*, oferecendo uma experiência mais atraente e envolvente para os utilizadores, permitindo-lhes interagir diretamente com o Chat GPT; quer seja numa conversa privada, ou então num chat público, através do comando: `!gpt {prompt desejado}`.

NOTA: O *user* com o nome "Chat GPT" deve estar criado na base de dados para serem utilizadas estas funcionalidades, pois é assim que ele é chamado no código desenvolvido.

Para além disto, é de realçar que o site não se limita a um grupo restrito, estando apto para qualquer utilizador. Através das funcionalidades de filtragem, cada utilizador pode escolher os *subreddits* que deseja ver, de acordo com as suas preferências.

De notar que o Serit.Inc é um *website* aberto a qualquer tema, permitindo que qualquer instituição possa utilizar o código da maneira que quiser, desta forma, demonstrando que o código é escalável e adaptável. Na confeção deste, foi importante sempre garantir uma integridade e modularidade dos ficheiros, sendo todos estes manuseados em pastas separadas, recorrendo a *templates* (em formato HTML 5.0) e *static* (CSS e javascript, sem informações *inline* no HTML), de acordo com os últimos standards W3C. Para além disso, todos os ficheiros herdam de um determinado estilo CSS, evitando repetição de código entre os ficheiros. Este processo facilita caso alguma identidade queira realizar alguma alteração ao código do site (alterar cor, tipo de letra, etc…). Ainda, de forma a garantir o encapsulamento do código e a segurança do código fonte, é separada a validação das *views* e o HTML que o utilizador vê, para que este não obtenha acesso à "*backend*" do *website*.

Aspetos relevantes do Código e recursos externos

Segurança e privacidade

✓ Google Captcha

A utilização do Captcha (Caixa de verificação v2) é relevante para proteger o site contra *bots* automatizados, garantindo a autenticidade dos utilizadores. Ele previne atividades maliciosas, como registos automáticos em massa e envio automático de formulários, melhorando a segurança e a integridade das interações no site.

NOTA: Ao correr o programa pode aparecer que falta fazer uma migração, no entanto se se aplicar a migração vai apagar o modelo CaptchaStore, responsável pela verificação.

✓ Google *AUTH*

A implementação da autenticação com contas Google em um projeto Django, usando a biblioteca django-allauth, oferece inúmeras vantagens. Para concretizar este objetivo, utilizou-se a *API* do *Google Cloud*. Em primeiro lugar, simplifica o acesso dos utilizadores, já que a maioria possui uma conta Google, eliminando a necessidade de criar *passwords* adicionais. Além disso, também facilita a partilha de informações de perfil, melhorando a experiência do utilizador, e, é compatível com várias plataformas, o que é vital para alcançar um público amplo. Assim, a integração com contas Google através do django-allauth simplifica a autenticação, promove a segurança e melhora a experiência do utilizador de forma significativa. Neste tópico, não temos a página de login customizada, já que por *default* estava apresentada sem qualquer estilo e caso quiséssemos proceder à sua customização, teríamos de descarregar a pasta *templates* da biblioteca e fazer o respetivo CSS, o que excederia o limite de 250 MB do zip para submissão.

Extras

✓ Chat GPT

Como já foi referido nos pontos inovadores, foi utilizada uma *API* do Chat GPT, onde tivemos de ter acesso de uma *key* para o funcionamento do mesmo.

✓ *Bootstraps*

Para mudar o estilo de algumas ações no projeto foram utilizadas ferramentas para estilização CSS, nomeadamente em fontes, que podem ser acessadas pelos seguintes links de *bootstrap*:

(I) https://cdnjs.com/libraries/font-awesome (algumas classes nas heranças do *index*)
(II) https://github.com/twbs/bootstrap/tree/main/dist/css (janela menor e botões do *login*)

Relativamente à página de registo do utilizador no *site*, decidimos exportar ficheiros de JavaScript e CSS para a seleção da nacionalidade. Desta forma, temos a certeza de que qualquer utilizador, de qualquer parte do mundo se sinta acolhido pelo s*ite*! O *link* de que foram retirados estes ficheiros foi este, inspirado no bootstrap do criador mojoaxel (*link* no nome).

✓ Whitenoise

Neste projeto foram desenvolvidos e personalizados *layouts* para as páginas de erro 404 e 500. Consequentemente, foi necessário colocar *DEBUG = False* nas definições do django e correr o servidor com o comando `python manage.py runserver --insecure`. Esta alteração nas definições originou a quebra dos "*media files"*, por parte do django, tendo sido necessário encontrar uma solução para este problema. Após inúmeras tentativas, a solução encontrada foi a ***seguinte.*** Esta sugere que criemos uma linha nos *settings* para servir os ficheiros *media* (SERVE_MEDIA_FILES) e a alteração dos urls raiz.