TarefaNest

Luiz Henrique Carneiro Carvalho

PROJETO TAREFANEST: Uma solução para gerenciar suas tarefas pelo computador e celular.

Feira de Santana, BA 2023

Sumário

1.Introdução	3
2.Ferramentas da página	4
3.Banco de Dados	7
3.1. Método Construtor	7
3.2. SQLite	7
3.2.1. Alter Table SQLite	8

1.Introdução:

Com a proposta de criar um aplicativo de anotação de tarefas, desenvolveu-se um projeto robusto utilizando o framework Django, uma ferramenta poderosa para construção de aplicativos web em Python. O aplicativo foi posteriormente hospedado na plataforma Heroku, proporcionando uma presença online acessível tanto por computadores quanto por dispositivos móveis.

A escolha do Django como o framework back-end ofereceu uma estrutura organizada e eficiente para gerenciar o lado servidor da aplicação. Utilizando o SQLite como banco de dados padrão, o projeto se beneficiou de uma solução leve e fácil de configurar, ideal para aplicações de pequeno a médio porte.

O layout responsivo foi implementado através do Bootstrap, uma biblioteca de código aberto que simplifica o design e a estilização do front-end. Isso assegura uma experiência consistente e amigável ao usuário, independentemente do dispositivo utilizado para acessar o aplicativo.

O armazenamento de dados e a interação do usuário são mediados por meio do Python, aproveitando a simplicidade e a versatilidade dessa linguagem de programação. A integração do Django com o SQLite e o uso do Python para manipulação de dados tornaram possível a criação, leitura, atualização e exclusão de tarefas de maneira eficaz.

Em resumo, o projeto incorpora tecnologias modernas como Django, Bootstrap, SQLite e Python, proporcionando uma solução flexível e eficaz para a gestão de tarefas. A decisão de hospedar na Heroku garante uma presença online estável e acessível a usuários de diferentes dispositivos, cumprindo assim os requisitos de um aplicativo de anotação de tarefas versátil e prático.

2. Ferramentas da página:

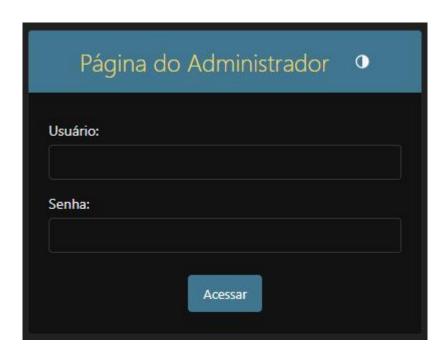


Na página de Criar tarefa, é possível escrever o título, descrição, data e status da tarefa. E quando clicar no botão de criar, ela é armazenada no banco de dados do site.

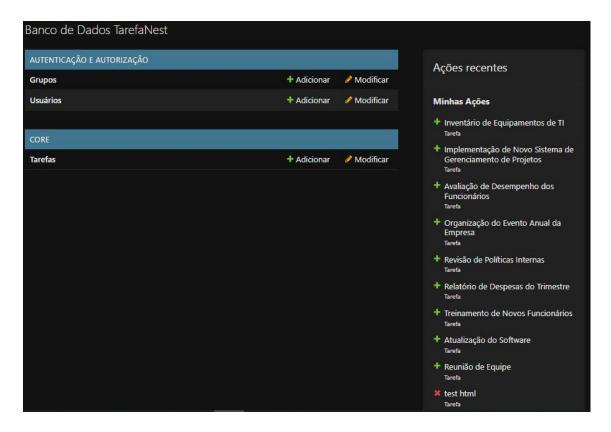
Tarefas Disponíveis

Título	Descrição	Data	Status	Ação
Limpar escritório	Fazer a limpeza dos escritórios que estão no andar 17	16 de Janeiro de 2024	Pendente	Apagar
Reunião de Equipe	Discutir metas trimestrais e atribuir responsabilidades para cada membro da equipe.	23 de Fevereiro de 2024	Concluída	Apagar
Atualização do Software	Atualizar o sistema operacional em todos os computadores da empresa para a versão mais recente.	5 de Março de 2024	Pendente	Apagar
Treinamento de Novos Funcionários	Realizar o treinamento inicial para os novos funcionários recém-contratados, abordando procedimentos internos e cultura organizacional.	12 de Abril de 2024	Executando	Apagar
Relatório de Despesas do Trimestre	Compilar e analisar as despesas da empresa no último trimestre para apresentação à diretoria.	30 de Abril de 2024	Concluída	Apagar
Revisão de Políticas Internas	Revisar e atualizar as políticas internas da empresa para garantir conformidade com as leis atuais e as melhores práticas do setor.	15 de Maio de 2024	Executando	Apagar
Organização do Evento Anual da Empresa	Planejar e coordenar todos os aspectos do evento anual da empresa, incluindo local, catering, palestrantes e logística.	8 de Junho de 2024	Pendente	Apagar

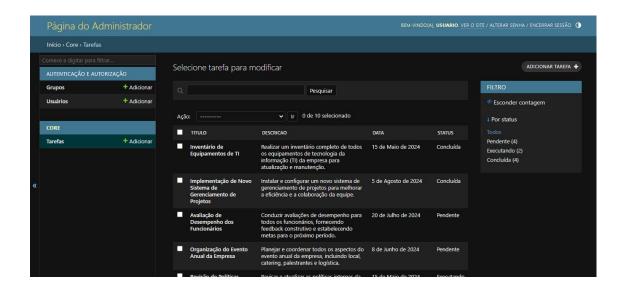
Na página de Tarefas Disponíveis, já é possível ver todo o banco de dados do site, tendo a opção de alterar status e apagar tarefa já na tabela.



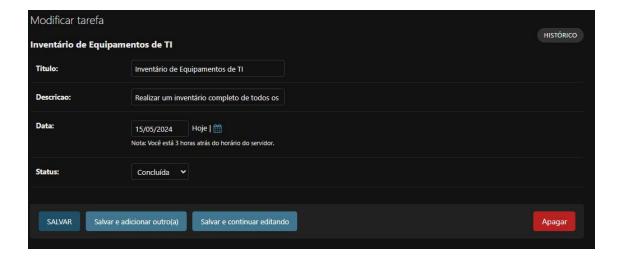
Seguindo pela página do administrador, para acessar ela é necessário ter uma conta admin com os campos de usuário e senha (para essa demonstração tanto o nome como senha do perfil são "usuario").



Na página inicial do admin, além das funções padrões do django como criar grupos e usuários, você pode ver o histórico de alterações e acessar o banco de tarefas.



Acessando a página de tarefas, você pode ver todas as tarefas disponíveis, filtrar o status para o que você quiser e poder modificar outros atributos que não são possíveis na página comum.



Sessão onde você modifica qualquer atributo da tarefa.

3.Banco de Dados

Django (método construtor):

```
class Tarefa(models.Model):
  STATUS_CHOICES = [
    ('pendente', 'Pendente'),
    ('executando', 'Executando'),
    ('concluida', 'Concluída'),
  1
  titulo = models.CharField(max length=200)
  descricao = models.CharField(max length=5000)
  data = models.DateField()
  status = models.CharField(max length=20, choices=STATUS CHOICES,
default='pendente')
  def str (self):
    return self.titulo
SQLite:
- Criando:
BEGIN;
-- Create model Tarefa
CREATE TABLE "core_tarefa" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT,
 "titulo" varchar(200) NOT NULL,
"data" date NOT NULL,
"status" varchar(20) NOT NULL);
COMMIT;
```

- Fazendo Alter Table: BEGIN; -- Add field descricao to tarefa - CREATE TABLE "new__core_tarefa" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "descricao" varchar(5000) NOT NULL, "titulo" varchar(200) NOT NULL, "data" date NOT NULL, "status" varchar(20) NOT NULL); INSERT INTO "new__core_tarefa" ("id", "titulo", "data", "status", "descricao") SELECT "id", "titulo", "data", "status", 'descricao' FROM "core_tarefa"; DROP TABLE "core_tarefa"; ALTER TABLE "new__core_tarefa" RENAME TO "core_tarefa";

COMMIT;