IMPORTANTE

Focar nos Argumentos e justificativas que demonstrem a necessidade e adesão do portal por parte das Prefeituras e Usuário visando a apresentação do Projeto.

Lembrar de argumentar com dados estatísticos os benefícios para a cidade e sua população

O portal vai oferecer os dados relacionados a vacinação e castração divididos por bairro para ajudar a direcionar as campanhas publicas relacionadas a esses tópicos.

Demonstrar o papel do município durantes as campanhas publicar de vacinação e castração devem exigir a adesão do chip custeados pela própria prefeitura e o cadastro do animal e proprietário sendo feitos na hora

Papel do Veterinário oferecendo serviço de chipagem e cadastro em suas clínicas facilitando a identificação e controle dos dados dos animais

Precisamos fazer:

Deixar Documentação de lado e focar exclusivamente no código e apresentação do projeto

Redirecionar o trabalho para a visão da Autoridade Pública, e Veterinários/parceiros que iram ficar responsáveis pelo cadastro dos animais e de seus tutores

Criar soluções de incentivo para os usuários cadastrados (Descontos com Clínicas e Petshops)

Ficha Única do Animal (vai poder ser acessada por todos os veterinários) ajudando a tornar os dados acessíveis e centralizados.

Criar Dashboard (Tela de usuários) com as funcionalidades correspondentes para cada usuário:

Dashboard Administrador do Portal(Cadastrar e dar nível de acesso para: Autoridades/Veterinário, Editar dados Cadastrais, Gerenciar Comunidade(Postagens de campanhas e promoções), Gerenciar todos os usuários, Definir filtros para exibição dos dados do portal)

Dashboard Autoridades (cadastro de pessoas e animais, busca de animais, ler e responder mensagens de Denúncias, botão de atalho para estatísticas)

Dashboard Veterinários (cadastro de pessoas e animais, busca e visualização de animais)

Dashboard Usuário Padrão (Meus animais cadastrados, meus Dados, Minhas mensagens: Animais Achados/Perdidos) apenas alguns campos serão editáveis tanto dos dados do usuário quanto do pet: endereço, telefone, vacinas etc. Pensar em maneiras seguras de fazer a ligação entre animal e seu dono.

Fazer a ligação e configuração do backend utilizando o MongoDB (Lembrar da a lógica 1 para muitos (Tutor/Animais)

Fazer a ficha do Animal recuperando todos os dados do banco

Fazer todo Front-End e Back-End das funcionalidades de todos os Dasboard

Ajustar o front da pagina comunidade.html e fazer o backend com suas funcionalidades

Ná página Informações.html Fazer ela exibir o mapa da cidade dividido por bairros mostrando a quantidade animais vacinados e castrados em realção a quantidade total de animais

Ajustar o front das paginas de cadastro para deixar de forma individual sem relação entre elas deixando ela reutilizável para todos (esconder certos campos para os usuários normais)

Criar a possibilidade de vincular o animal ao usuario

Ajustar o front de todas as paginas para melhor exibição

Criar mecanismo de Busca de Animais Com Filtros(Nº do Chip de rastreamento, CPF do Tutor, Nº Ficha de Cadastro no sistema)

Material de apoio : Para criar o backend para o Dashboard mencionado, você precisará de um sistema que gerencie usuários, autoridades, veterinários, e as diferentes funcionalidades associadas a cada tipo de usuário. Aqui está uma explicação geral das funcionalidades e a lógica por trás do backend:

**Cadastro e Níveis de Acesso:**

**Lógica:** Implemente um sistema de autenticação para o cadastro de usuários. Armazene as informações de login de forma segura, e associe a cada usuário um nível de acesso (administrador, autoridade, veterinário, usuário padrão).

**Funcionalidades:**

Cadastro de Administradores com a capacidade de conceder níveis de acesso a Autoridades/Veterinários.

**Editar Dados Cadastrais:**

**Lógica:** Permita que administradores e usuários autorizados editem seus dados pessoais.

**Funcionalidades:**

Editar dados cadastrais para administradores, autoridades, veterinários e usuários padrão.

**Gerenciar Comunidade:**

**Lógica:** Desenvolva um sistema de postagem para campanhas e promoções e implemente um sistema de gerenciamento dessas postagens.

**Funcionalidades:**

Administradores podem criar, editar e excluir postagens.

Usuários podem visualizar postagens e interagir de acordo com seus níveis de permissão.

**Gerenciar Todos os Usuários:**

**Lógica:** Permita que administradores vejam e gerenciem todos os usuários do sistema.

**Funcionalidades:**

Listar todos os usuários cadastrados.

Permitir que administradores editem e excluam contas de usuário, se necessário.

**Definir Filtros para Exibição dos Dados:**

**Lógica:** Implemente filtros para personalizar a exibição dos dados conforme necessário.

**Funcionalidades:**

Permitir que administradores definam filtros para visualizar dados específicos do portal.

**Dashboard Autoridades:**

**Lógica:** Autoridades têm permissões específicas para lidar com casos, estatísticas e cadastros.

**Funcionalidades:**

Cadastro de pessoas e animais.

Busca de animais.

Leitura e resposta de mensagens de denúncias.

Botão de atalho para estatísticas.

**Dashboard Veterinários:**

**Lógica:** Veterinários têm permissões específicas relacionadas ao cadastro e visualização de animais.

**Funcionalidades:**

Cadastro de pessoas e animais.

Busca e visualização de animais.

**Dashboard Usuário Padrão:**

**Lógica:** Usuários padrão têm acesso limitado, principalmente relacionado a seus animais de estimação.

**Funcionalidades:**

Visualizar seus animais cadastrados.

Editar alguns campos dos dados do usuário e do pet.

Visualizar mensagens relacionadas a animais achados/perdidos.

**Segurança da Ligação entre Animal e Dono:**

**Lógica:** Garanta que a relação entre o animal e seu dono seja segura e protegida.

**Funcionalidades:**

Implementar um sistema de identificação seguro para vincular animais a seus proprietários.

Garantir que apenas usuários autorizados tenham acesso a essas informações.

**ChekList:**

**Configuração do Ambiente:**

Configurar um ambiente de desenvolvimento local com PHP e um servidor web (por exemplo, Apache ou Nginx).

Configurar um banco de dados (por exemplo, MySQL, PostgreSQL).

**Backend (PHP):**

Configurar um framework ou biblioteca para facilitar o desenvolvimento (por exemplo, Slim, Laravel).

Implementar o sistema de autenticação para usuários (registro, login, logout).

Criar um modelo de dados para usuários, autoridades, veterinários, animais, etc.

**Frontend (HTML/CSS/JS):**

Configurar a estrutura básica do frontend (páginas HTML, folhas de estilo CSS).

Implementar páginas de registro, login e dashboard para cada tipo de usuário.

**Cadastro e Níveis de Acesso:**

Criar formulário de registro no frontend.

Implementar endpoint no backend para processar o registro.

Associar níveis de acesso aos usuários.

**Editar Dados Cadastrais:**

Criar formulário de edição de perfil no frontend.

Implementar endpoint no backend para processar a edição do perfil.

**Gerenciar Comunidade:**

Implementar sistema de postagem no frontend.

Criar endpoint no backend para gerenciar postagens (criar, editar, excluir).

**Gerenciar Todos os Usuários:**

Implementar página no frontend para listar todos os usuários (apenas para administradores).

Criar endpoint no backend para obter a lista de usuários.

**Definir Filtros para Exibição dos Dados:**

Adicionar opções de filtro na interface do administrador.

Implementar lógica no backend para aplicar os filtros.

**Dashboard Autoridades:**

Implementar funcionalidades específicas para autoridades no frontend.

Criar endpoints no backend para lidar com operações específicas de autoridades.

**Dashboard Veterinários:**

Implementar funcionalidades específicas para veterinários no frontend.

Criar endpoints no backend para lidar com operações específicas de veterinários.

**Dashboard Usuário Padrão:**

Implementar funcionalidades específicas para usuários padrão no frontend.

Criar endpoints no backend para lidar com operações específicas de usuários padrão.

**Segurança da Ligação entre Animal e Dono:**

Implementar um sistema seguro de vinculação entre animais e usuários.

Garantir que apenas usuários autorizados tenham acesso a essas informações.

**Testes e Validação:**

Realizar testes de unidade e integração.

Validar a segurança do sistema (injeção de SQL, XSS, CSRF).

Testar a integração entre frontend e backend.

**Documentação:**

Documentar o código (comentários, README).

Descrever a arquitetura do sistema e como configurá-lo.

**Deploy:**

Preparar o ambiente de produção.

Reali