Apresentação e defesa do projeto

O site foi feito para ajudar quem tem dificuldade de mobilidade urbana, seja por deficiência, idade ou por ser estrangeiro. Ele mostra locais acessíveis, seus horários, descrições e tipos de acessibilidade, e ainda permite que os próprios usuários publiquem e avaliem esses lugares. Além disso, profissionais podem oferecer serviços de apoio, com valores, horários e pagamentos online.

Em um cenário de aplicação, o site funciona como um guia prático para quem precisa se locomover com segurança e conforto na cidade. Por exemplo, uma pessoa idosa que quer sair para um compromisso pode consultar os locais próximos, verificar se são acessíveis e ainda contar com serviços de apoio, como acompanhantes ou transporte especializado, tudo registrado e disponível para agendamento. Já profissionais e estabelecimentos ganham visibilidade e podem receber avaliações, garantindo maior confiança e transparência para todos.

A viabilidade do projeto é evidente porque ele utiliza recursos já existentes, como cadastro de usuários, sistema de avaliação, agendamento e pagamento online, aplicados de forma a resolver um problema real da população. A tecnologia necessária é simples, e o modelo de participação comunitária permite que a plataforma cresça e se atualize constantemente, sem depender de grandes investimentos.

Em uma simulação de implementação em cenário real, o site poderia ser usado inicialmente em uma cidade piloto. Usuários cadastrados publicariam locais acessíveis e profissionais cadastrariam seus serviços. Cada usuário poderia buscar serviços e locais, agendar atendimentos e fazer pagamentos pelo próprio sistema. À medida que mais pessoas utilizassem a plataforma, os dados se tornariam cada vez mais completos e confiáveis, criando uma rede de inclusão e suporte que conecta cidadãos, profissionais e estabelecimentos, tornando a cidade mais acessível de forma prática e eficiente.

Relatório da Arquitetura – Back-End

Quando comecei a montar o back-end do meu projeto, a minha ideia principal foi deixar tudo organizado e fácil de mexer depois. Eu usei Python para construir a parte da lógica, dá pra testar rápido se as coisas estão funcionando como eu quero, tipo cadastro, login, agendamentos e essas partes que precisam de regras.

Como eu pretendo colocar o sistema no Firebase, já fui pensando nisso enquanto fazia o código. Então, tudo que eu fiz em Python agora funciona como uma base, meio que um rascunho para o que mais tarde vai ficar no Firebase, onde os dados vão realmente ser guardados.

O Firebase vai ficar responsável pela parte de contas dos usuários e pelos dados do sistema, como serviços, horários, publicações e tudo mais. Mas antes de jogar tudo pra lá, eu quis organizar primeiro em Python, porque assim eu consigo testar tudo, ver onde precisa arrumar e entender melhor o fluxo do sistema.

No geral eu separei as funções direitinho, tipo: cadastrar, fazer login, publicar, avaliar, agendar… cada coisa em sua parte. Isso deixa tudo mais claro pra mim e também facilita na hora de passar isso para o Firebase.

Então basicamente o back-end foi feito assim:
primeiro eu montei toda a lógica em Python, de um jeito simples e fácil de entender, e depois eu vou levar isso pro Firebase, onde ele vai funcionar de verdade na parte online.

Relatório de Dados – Firebase

Quando eu fui montar a parte de armazenamento do projeto, a minha intenção era começar passando só o cadastro dos usuários para o Firebase, pra depois ir colocando o resto.

Primeiro, eu configurei o Firestore e criei a coleção chamada **"cadastro"**, que seria onde ficariam guardados os dados de cada usuário cadastrado pelo sistema. A ideia era: toda vez que alguém preenchesse o formulário no HTML, o Python ia validar tudo e depois mandar as informações pro Firebase.No código, eu preparei tudo certinho: conexão com o Firebase, validação de nome, telefone, email e senha, e depois a parte de enviar o documento para o Firestore, o objetivo era só fazer o cadastro funcionar. Quando tentei rodar o código Python com o Firebase, ele até conectou direitinho, mas na hora de adicionar o usuário deu erro. O problema era que a chave JSON do Firebase não estava sendo lida corretamente pelo PyWebView, e eu fiquei um bom tempo tentando resolver. A ideia era: o HTML chama o Python, o Python valida e manda pro Firebase, mas justamente nessa etapa final deu o problema. Eu tentei fazer funcionar pelo menos o cadastro no Firebase para já ter algo salvo lá, mas não consegui resolver esse erro a tempo.