This is a prompt to continue building the site. The previous additions are great but I hadn't finished my prompt: While the previous code works, I need to be able to sign up and then sign in so that I can be able to view the profile I created, update my profile in the settings, add team squad, etc. Right now that is not possible. It's only forms that lead to nowhere. We need to sort this out so we can check the progress of our site. And here are more prompts.

1. Donor: Anyone can be a donor(e.g individual, team, organization, business, shop, etc) so because of this, their profile view will have to reflect who they are ( a business, team, org, etc). * Will have a profile with settings tailored to their user type:

- must be able to update their banking details as well as have multiple cards, also with an ability to delete their cards.
- Must be able to update their names, location, profession, business type
- Must be able to upload their profile picture(business logo/personal picture if an individual)
- Must be able to delete their profile if they no longer need it.
- Can choose for their profile to be anonymous when they've made a donation
- On their profile, they must be able to see their donations data, which will include amount or items(for example, teams may donate balls and shops may donate goal posts) and also the teams they've donated to. The teams on the list must be clickable so that the donor can be able to see the team's profile without having to redirect.
- In order to have these features functional, create all the necessary files including payment gateways or stripe integration, card forms, etc.
- You might find these useful: To design the donor profile, focusing on security and ease of use is key, especially around capturing banking details. You can take a few steps to ensure smooth functionality:

  1. **Registration Process**: Start by collecting basic information from donors, like name, email, phone number, and address. This helps with tracking donations and providing receipts. 2.

  **Payment Options**: Offer multiple payment methods, such as credit/debit cards, bank transfers, and even PayPal or other secure platforms.

  3. **Secure Card Details**: Implement a form with card input fields that are integrated with a payment gateway (like Stripe, PayPal, or Square) to process payments. These platforms offer security features like tokenization (so no sensitive data is stored on your servers) and are PCI-DSS compliant, ensuring a high level of security.

  4. **Two-Factor Authentication (2FA)**: You could set up SMS-based 2FA for transaction confirmations, where donors receive a one-time password (OTP) to finalize payments, adding an extra layer of fraud prevention.

  5. **In-Kind Donations**: For non-financial donations (like goalposts, kits, etc.), you could have a form where donors can specify the type and quantity of goods they want to donate. Linking this to a logistics partner for delivery coordination can be useful.

* Here’s a brief look at the **low-level implementation process** for integrating Stripe into the platform:

1. **Install Stripe SDK**: Install Stripe’s client-side SDK (e.g., Stripe.js for JavaScript) into your web application to securely capture card details. This ensures card information never touches your server, maintaining PCI compliance.

2. **Create a Payment Form**: Build a form where donors can input their card details (card number, expiration date, and CVV). Use Stripe Elements to customize the card input fields securely.

3. **Tokenize the Card Details**: When the donor submits the form, Stripe tokenizes the card data (turning it into a secure, single-use token). This token can now be used instead of raw card details.

4. **Send Token to Server**: Pass the token to your server (via an API call) where you can create a payment request to Stripe's backend.

5. **Process the Payment**: On the backend, use the Stripe API to process the payment by sending the token, amount, and any additional details like currency.

6. **Handle Payment Response**: Stripe will respond with success or failure. If successful, update the donor’s account, send confirmation emails, etc. If unsuccessful, display appropriate error messages.

7. **Security Measures**: Optionally, integrate 3D Secure or two-factor authentication (provided by Stripe) for additional fraud protection during the payment process.