Steps to build the App

1. Set up your development environment: You need to have Python and Flask installed on your computer to build the application. You can use virtual environments to manage dependencies for your project.
2. Define the database schema: Determine the tables and fields you need to store the information about the children and their parents, education, and home situation, as well as their goals and challenges. You can use SQLAlchemy, a popular Object Relational Mapper (ORM) library, to interact with your database.
3. Create the Flask application: Start by creating a Flask application and defining the routes that the user will interact with. You can also define the template pages to render the information stored in the database.
4. Integrate the database: Connect your Flask application to the database using SQLAlchemy and create the tables you defined in step 2. You can also add sample data to your database to test your application.
5. Add CRUD functionality: Implement the Create, Read, Update, and Delete (CRUD) operations for the children and their information in the database. This will allow you to add, view, edit, and delete records from the database.
6. Add authentication and authorization: Add authentication and authorization to the application to ensure that only authorized users can access and modify the data. You can use the Flask-Login library to handle authentication and the Flask-Admin library to handle authorization.
7. Test and Deploy: Test your application thoroughly and make any necessary changes. Finally, you can deploy your application on a web server, such as Heroku or AWS, to make it available to users.

2. Define the database schema:

* determining the structure of the data
* the relationships between different pieces of data

3. Create the Flask application:

* Create a new Flask project: Start by creating a new directory for your project and creating a main Python script that will serve as the entry point for your application.
* Define the routes: The routes define the different pages that the user will be able to access and interact with in your application. For example, you might have routes for a home page, a page to view the information about a specific child, and a page to add new children to the database. You can define these routes using the @app.route decorator in Flask.
* Create templates: Templates define the HTML pages that will be rendered in response to user requests. You can use the Jinja2 template engine in Flask to create these templates and pass data from your application to the templates to be displayed to the user.
* Set up the main application file: In the main application file, you will import the Flask library, create a new Flask application instance, and define the routes and templates you created in the previous steps. You will also configure any other settings for your application, such as the secret key for security.
* Test the application: Once you have defined the routes and templates, you can test your application locally to make sure everything is working as expected. You can start the application by running the main Python script you created, and then accessing the routes in your web browser.