Task 35: Schema Reference

Problem Statement:

Your task is to create an Express.js application that demonstrates referencing one Mongoose schema to another. Additionally, create a React frontend to interact with this backend.

Problem Description:

In this task, you will set up a MongoDB database connection using Mongoose and define two schemas with a reference from one to the other. You will implement routes to create and retrieve related data. You will also create a React frontend to interact with these routes.

Your Express.js application should include the following features:

1. Connect to MongoDB: Use Mongoose to connect to a MongoDB database.
2. Define Schemas:
   - User Schema: Schema for storing user data with fields for name and email.
   - Post Schema: Schema for storing post data with fields for title, content, and a reference to the User schema.
3. Create Routes:
   - POST /users: Add a new user to the database.
   - POST /posts: Add a new post linked to a user.
   - GET /posts: Retrieve all posts with user information populated.

Your React application should include the following features:

- Forms for Input: Forms to collect user and post data.
- Submit Data: Functionality to send POST requests to add users and posts.
- Display Data: Functionality to fetch and display the list of posts with user information.

Submission Guidelines:

- **1. Deploy both frontend and backend on vercel**
- **2. Upload both frontend and backend files on github.**
- **make a readme file in github frontend repo and paste the links of deployed frontend , deployed backend and backend github repo.**
- **Add ScreenShots from mongodb in Readme file.**
- **Submit the Frontend repo link in Dashboard**
- **Example frontend repo link: https://github.com/aakashk1905/DummyFrontEnd**