Hello everyone! My name is Kate, and today I want to share my project - an online platform that helps connect patients with psychologists. As an immigrant myself, I understand how hard it can be to find mental health support when you need it most.

**The Problem**

When I first moved to a new country, I felt very alone and stressed. Finding a psychologist was difficult and confusing. Many immigrants face the same problems:

- Language barriers

- Don't know where to find help

- Complicated booking systems

- Long waiting times

**My Solution**

I built a secure web platform that makes it easy to:

- Find available psychologists

- Book appointments online

- Manage your mental health journey

**I used modern technologies to build this platform**:

- Backend: Node.js with Express

- Database: MongoDB for secure data storage

- Frontend: Clean and simple HTML/JS interface

- Security: JWT tokens for safe login

- Special Feature: Smart email system that connects existing patients with their records

I created a Node/Express application from scratch using the MongoDB database. I have two Mongoose data models. One of these is a User

So I implemented user registration and logon, authentication use JWT tokens.

**Let me start by showing you our main page**. As you can see, it's clean and simple to use. Just like websites we use every day, but designed specifically for mental health support.

**Let's see how it works for a patient:**

1. **Registration Process**

"First, I'll show you how easy it is to register as a new patient:

- Click 'Sign Up' here

- Fill in basic information

- The system automatically checks if you're already in our database

Patients can also:

- View all their upcoming appointments

**Now, let's switch to a psychologist's account**:

- Different dashboard - see how it shows their full schedule?

- They can mark their available hours

- Review patient appointments

- Manage their calendar

**Implement access control logic in your controllers**, so that one user can’t access another user’s data

**Include appropriate notifications** to the user.

**Implement error handling middleware** so that all exceptions and error conditions are handled and so that the user gets user friendly messages for each event.

**Non-CRUD realized sorting**

**Include security protections** for application.

**Realized deployment.**

**Implemented Swagger to document the API**

## **Special Features**

**"Here are some cool features I built:**

1. **Smart Email System**

"Watch this:

- When I type an email address

- The system automatically checks our database

- If the patient exists, it fills in their information

- This saves time and prevents errors

2**. Error Handling**

**"Let me show you what happens if something goes wrong:**

- If I try to book a past date - see this friendly message?

- Everything is in simple, clear language

**## Future Updates**

"Before I finish, let me quickly show you what's coming next:

- This button will be for donation

I want to add more features:

- Multiple languages

- More booking options

Then, I want to share something interesting about my learning journey:

I took two classes and built two apps. It was possible to make just one, but I decided to learn how to use JavaScript without React, from scratch. It was a challenge, but I realized it was beneficial for my learning process. Now, I truly appreciate the benefits of React.

Building projects in different ways helped me understand:

- How much easier React makes web development

- Why React was created in the first place

- How event handling is simpler in React

- The real value of modern web frameworks

**I really love React.**

**Now, let me show you my Todo List application:**

1. User Interface: Let's start with our home page:

   - Clean, beautiful design with modern CSS

   - Smart sidebar that opens and closes with one click

   - Notice how easy navigate between pages

- **Let's navigate to our main Todo page**

2. **Task Management:**

**Let me show you how easy it is to add a new task**:

- Click the 'Add Task' button

- Type in a task title 'Meeting with Team'

- **Watch the validation in action**:

  \* If I try to submit empty - see this friendly error?

  \* If title is valid - task appears instantly

- **Notice how it's immediately saved to Airtable** Everything updates in real-time

**Now let's manage our tasks**:

- See these task cards? Each has **several options**:

  \* Check this checkbox to mark as complete

  \* Watch how the style changes for completed tasks

  \* Click edit to modify the title

  \* Use delete button to remove tasks

- **Everything updates in real-time**

   - Update tasks with one click

   - Mark tasks as complete or incomplete

Delete tasks when needed

   - **Sorting Features**

"Let me demonstrate our sorting capabilities:

- Click here to sort by title:

  \* See how tasks reorder alphabetically?

  \* Click again for reverse order

- Now let's sort by date:

  \* Newest first

  \* Click again for oldest first

- Notice how smooth these transitions are

4. **Professional Features**:   **Technical Highlights**

"Behind this simple interface:

- React Router handling navigation

- Airtable API for data storage

- Complete CRUD operations

- Real-time updates

- Validation and error handling

**## What Makes This Project Special**

"This project is special because:- It's simple but powerful- Easy to use but full-featured- Beautiful but practical- Shows what I learned about React

Thank you for letting me share both my learning journey and my project. My experience with both pure JavaScript and React made me a better developer, and this Todo application shows what I learned.