**CRUD Exercise:**

1. **Start a new Next .js 14 project** (App Router, TypeScript). Do **not** add Prisma or any database. Use tailwindcss for styling.
2. **The website should be fully translated in arabic and english using next-intl. link:** [**https://next-intl.dev/docs/getting-started/app-router/with-i18n-routing**](https://next-intl.dev/docs/getting-started/app-router/with-i18n-routing)
3. **Links to use:**
   * **Get users:** [**https://dummyjson.com/users**](https://dummyjson.com/users)
   * **Documentation:** [**https://dummyjson.com/docs/users**](https://dummyjson.com/docs/users)
4. **Use the DummyJSON Users API**
   * List → GET https://dummyjson.com/users?limit={limit}&skip={skip}
   * Create → POST https://dummyjson.com/users/add
   * Update → PUT https://dummyjson.com/users/{id}
   * Delete → DELETE https://dummyjson.com/users/{id}
5. **Create a paginated table page.**
   * Fetch users and show 10 per page.
   * Table columns **exactly**:
     1. **Name** = firstName + " " + lastName
     2. **Email** = email
     3. **Lat** = address.coordinates.lat
     4. **Lng** = address.coordinates.lng
     5. **Card Expire** = bank.cardExpire
     6. **Card Number** = bank.cardNumber
   * Add one **Create User** button above the table.
   * Add two action buttons in each row: **Update** and **Delete**.
6. **Create User flow.**
   * Clicking **Create User** opens a form page or modal.
   * Submit the form with POST /users/add.
   * After the request, return to the table and show the new user.
7. **Update User flow.**
   * Clicking **Update** on a row opens a form with that user’s data.
   * **Not all fields are editable**—disable the fields you do not want changed.
   * Send changes with PUT /users/{id}.
   * On success, return to the table and refresh that row.
8. **Delete User flow.**
   * Clicking **Delete** sends DELETE /users/{id}.
   * After the request finishes, remove that row from the table.
9. **User Details page.**
   * Double-clicking a row opens /users/{id}
   * Show **all six table fields** **plus** every property inside the user’s company object (department, name, title and the full company address).

Example image of the table design:

Ar:

A screenshot of a computer

Description automatically generated

En:

A screenshot of a computer

Description automatically generated

Colors to use:

const colors = {

info: {

main: "#A9C7BF",

light: "#D5E1B5",

dark: "#2A6C57",

},

warning: {

light: "#ffbf00",

main: "#ef6c00",

},

secondary: {

main: "#9d9d9d",

dark: "#414141",

light: "#ededed",

},

success: {

main: "#35795D",

dark: "#295e48",

light: "#B4DD4F",

},

error: {

main: "#FF1744",

},

text: {

light: "#dadada",

dark: "#555555",

},

};

export default colors;

Use those colors in tailwind.congif.ts

extend: {

colors: {

info: {

main: colors.info.main,

light: colors.info.light,

dark: colors.info.dark,

},

warning: {

light: colors.warning.light,

main: colors.warning.main,

},

secondary: {

main: colors.secondary.main,

dark: colors.secondary.dark,

light: colors.secondary.light,

},

success: {

main: colors.success.main,

dark: colors.success.dark,

light: colors.success.light,

},

error: {

main: colors.error.main,

},

text: {

light: colors.text.light,

dark: colors.text.dark,

},

},

},