



*Resource Software Solution*

**ReactJS**

**On the Job Training**

# Dự án “Personal Task Management App” (ReactJS + Supabase)

## 1. Ý tưởng dự án: "Personal Task Management App"

### Giá trị thực tiễn:

- Quản lý task cá nhân, chia nhỏ task, deadline, ưu tiên, trạng thái, checklist, nhắc nhở, bộ lọc, đăng ký đăng nhập.
- Học được đầy đủ quy trình dự án thực tế (code, test, CI/CD, tích hợp cloud backend – Supabase).

## 2. Tech Stack

- **Frontend:** ReactJS (Vite/CRA), Redux Toolkit, Axios, TailwindCSS, React Hook Form, React Router v6, Jest & React Testing Library
- **Backend:** Supabase (Auth, Database, REST API, Realtime)
- **Deploy:** Vercel
- **CI/CD:** GitHub Actions (bonus), Vercel auto deployment

## 3. Tài liệu đặc tả dự án

### 3.1. Yêu cầu nghiệp vụ (BRD – Business Requirement Document)

#### Mục tiêu:

- Người dùng đăng ký/đăng nhập
- Quản lý danh sách task (CRUD)
- Mỗi task có: tên, mô tả, deadline, trạng thái, độ ưu tiên, checklist con, file đính kèm (tùy chọn)
- Bộ lọc: trạng thái, deadline, ưu tiên, tìm kiếm

- Nhắc nhở (optional: notification hoặc chỉ highlight màu)
- Xem dashboard thống kê (tổng task, task hoàn thành, task trễ)
- Quản lý profile cá nhân

### 3.2. Đặc tả chức năng (FRD – Functional Requirement Document)

Mã chức năng	Tên chức năng	Mô tả
FR01	Đăng ký tài khoản	Đăng ký bằng email, mật khẩu (Supabase Auth)
FR02	Đăng nhập/đăng xuất	Đăng nhập/Đăng xuất (JWT – Supabase Auth)
FR03	Tạo mới task	Form tạo task, nhập tên, mô tả, deadline, ưu tiên, checklist
FR04	Xem danh sách task	Hiển thị tasks theo user, có phân trang hoặc infinite scroll
FR05	Cập nhật, xoá task	Sửa/xoá task trong list, xác nhận khi xoá
FR06	Bộ lọc & tìm kiếm	Lọc theo trạng thái, deadline, ưu tiên, tìm kiếm theo tên
FR07	Thống kê task	Dashboard: số task, hoàn thành, trễ deadline
FR08	Cập nhật hồ sơ cá nhân	Cập nhật thông tin cá nhân
FR09	Upload file đính kèm (tùy chọn)	Upload file lên Supabase Storage, lưu link file vào task
FR10	Nhắc nhở/highlight task	Highlight task sắp/đã trễ, hiển thị badge số ngày còn lại

### 3.3. API Spec (Supabase Table Structure đề xuất)

- **Table: users**
  - id (uuid, PK)
  - email
  - full\_name
  - created\_at
- **Table: tasks**
  - id (uuid, PK)

- user\_id (FK)
- title
- description
- deadline (timestamp)
- priority (enum: low/medium/high)
- status (enum: todo/in\_progress/done)
- checklist (json)
- attachment\_url (text)
- created\_at, updated\_at

### 3.4. Test Case (sample)

Tên test	Input	Expected Output	Priority
Đăng ký thành công	Email, Pass hợp lệ	Trả về userId, chuyển sang login	High
Đăng nhập với user chưa tồn tại	Email, Pass bất kỳ	Báo lỗi, không login	High
Tạo task không nhập tên	Trống title	Báo lỗi "Không được bỏ trống"	Medium
Sửa task với deadline đã qua	Chọn deadline < now	Báo lỗi hoặc highlight	Medium
Bộ lọc task theo trạng thái	Chọn "Done"	Hiện task có status "Done"	High
Upload file đính kèm	File ảnh/pdf	Lưu url vào task, tải lên thành công	Low

### 4. Roadmap triển khai (8 tuần/1 người)

Tuần	Giai đoạn	Nhiệm vụ
1	Khởi tạo	- Setup project React + Tailwind, Git, Vercel - Tạo repo, chuẩn hóa convention
2	Auth	- Tích hợp Supabase Auth: Đăng ký, Đăng nhập/Đăng xuất - Route bảo vệ, test case Auth
3	CRUD Task	- Tạo giao diện list, form thêm/sửa task - Kết nối API Supabase, test CRUD task
4	Checklist & Upload	- Thêm chức năng checklist, upload file - Validate form với React Hook Form
5	Bộ lọc, tìm kiếm	- Làm UI bộ lọc, tìm kiếm, dashboard thống kê
6	Profile, UI	- Trang profile cá nhân, tối ưu UX/UI bằng Tailwind
7	Testing	- Viết Unit test cho Redux, component, form - Integration test (mock API) bằng Jest/RTL
8	Deploy, CI/CD	- Viết tài liệu, config deploy Vercel - Optional: CI/CD với GitHub Actions

## 5. CI/CD – Deploy Vercel

- Kết nối GitHub repo với Vercel (auto deploy mỗi lần push main)
- (Optional) Thêm GitHub Actions: lint, run test trước khi merge/push

## 6. Hướng dẫn sử dụng Supabase Backend

### 6.1. Đăng ký Supabase

- Đăng ký account tại [supabase.com](https://supabase.com)
- Tạo Project mới → Setup bảng **users**, **tasks** như trên

### 6.2. Tạo Table, Auth, Storage

- Sử dụng SQL Editor tạo bảng
- Bật Row Level Security cho bảng (chỉ user truy cập task của mình)
- Tạo policy:

```
1. -- Chỉ cho phép user truy cập task của chính mình
2. CREATE POLICY "Users can view their tasks" ON tasks
3. FOR SELECT USING (user_id = auth.uid());
```

- Lưu key API, URL để frontend gọi qua Axios

### 6.3. Tích hợp Frontend với Supabase

- Cài package:

```
1. npm install @supabase/supabase-js
```

- Khởi tạo client:

```
1. import { createClient } from '@supabase/supabase-js';  
2. const supabase = createClient('SUPABASE_URL', 'SUPABASE_ANON_KEY');
```

- Dùng Supabase Auth cho login, signup
- Gọi API CRUD task (hoặc dùng restful endpoint của Supabase)

## 7. Checklist kiến thức cần đạt (Fresher → Junior)

Kỹ năng cần đạt	Mục tiêu kiểm chứng trong project
React (hook, router, lifecycle)	Sử dụng functional component, custom hook
State management (Redux Toolkit)	Quản lý state toàn app, async thunk
Axios, API call, error handling	Giao tiếp Supabase, handle lỗi, loading
UI/UX TailwindCSS, responsive	Giao diện mobile-first, component reuse
Form validation (react-hook-form)	Validate, custom rules, feedback UI
Unit/Integration Test (Jest/RTL)	Test logic, test form, test API mock
Backend as a Service (Supabase)	Auth, database, storage, policy
CI/CD deploy Vercel	Tự động deploy, rollback, log error
Documentation, git convention	README, commit lint, PR flow



## 8. Gợi ý phân chia giai đoạn code/test/deploy

- **Giai đoạn code:**
  - Thiết kế UI từng màn
  - Tích hợp từng feature nhỏ, commit từng phần, code review
- **Giai đoạn test:**
  - Viết unit test cho function, component
  - Integration test với mock API
- **Giai đoạn deploy:**
  - Triển khai bản thử nghiệm, nhận feedback, fix bug
  - Hướng dẫn rollback khi deploy lỗi

## 9. Mẫu thư mục chuẩn ReactJS tạo tại nhánh main

1. src/
2. api/
3. supabaseClient.js
4. taskApi.js
5. userApi.js
6. app/
7. store.js
8. components/
9. TaskCard/
10. TaskForm/

11. FilterBar/
12. ...
13. features/
14. auth/
15. AuthSlice.js
16. LoginPage.jsx
17. RegisterPage.jsx
18. tasks/
19. TasksSlice.js
20. TasksPage.jsx
21. ...
22. hooks/
23. useAuth.js
24. useTask.js
25. pages/
26. Dashboard.jsx
27. Profile.jsx
28. ...
29. styles/
30. tailwind.css
31. tests/
32. components/
33. features/
34. utils/
35. constants.js
36. helpers.js

## 10. Bảng quy định validation các trường và thông báo lỗi (message)

### - Màn hình FR01 - Đăng ký tài khoản (Sign Up)

Field	Validation Rule	Error Message (Thông báo lỗi)
Email	- Bắt buộc (required)- Đúng định dạng email (regex)	- Vui lòng nhập email- Email không hợp lệ
Mật khẩu	- Bắt buộc (required)- Tối thiểu 8 ký tự- Có ít nhất 1 số- Có ít nhất 1 ký tự đặc biệt (nếu áp dụng)	- Vui lòng nhập mật khẩu- Mật khẩu tối thiểu 8 ký tự- Mật khẩu phải chứa số- Mật khẩu phải chứa ký tự đặc biệt
Nhập lại mật khẩu	- Bắt buộc (required)- Trùng với mật khẩu đã nhập phía trên	- Vui lòng nhập lại mật khẩu- Mật khẩu không khớp
Đồng ý điều khoản	- Checkbox phải được tick- Không được submit nếu chưa đồng ý	- Bạn cần đồng ý với Điều khoản & Chính sách để tiếp tục

### Chi tiết và lưu ý bổ sung

- **Nút “Đăng ký” sẽ:**
  - Bị disable nếu bất kỳ trường nào sai hoặc thiếu.
  - Hiện thị lỗi phía dưới trường nhập bị lỗi.
- **UX:**
  - Tab di chuyển giữa các trường, focus rõ ràng.
  - Thông báo lỗi chuyển sang màu đỏ, font nhỏ, ngay sát input.
- **Sau khi đăng ký thành công:**
  - Thông báo: “Đăng ký thành công! Vui lòng kiểm tra email để xác thực tài khoản.”
  - Hoặc chuyển hướng sang màn đăng nhập.

### Gợi ý kiểm tra thực tế (Test case phổ biến)

Trường hợp kiểm tra	Kết quả mong đợi
Để trống tất cả các trường và nhấn Đăng ký	Báo lỗi tất cả các trường (email, mật khẩu, nhập lại, đồng ý)
Email sai định dạng	Báo lỗi “Email không hợp lệ”
Mật khẩu dưới 8 ký tự	Báo lỗi “Mật khẩu tối thiểu 8 ký tự”
Mật khẩu thiếu số hoặc ký tự đặc biệt	Báo lỗi tương ứng nếu áp dụng
Nhập lại mật khẩu khác mật khẩu ở trên	Báo lỗi “Mật khẩu không khớp”
Không tick đồng ý điều khoản	Báo lỗi “Bạn cần đồng ý với Điều khoản & Chính sách...”
Tất cả trường hợp hợp lệ, tick đồng ý, nhấn Đăng ký	Đăng ký thành công, chuyển hướng/trả thông báo

#### - Màn hình FR02 - Đăng nhập tài khoản:

Field	Validation Rule	Error Message (Thông báo lỗi)
Email	- Bắt buộc (required)- Đúng định dạng email (regex)	- Vui lòng nhập email- Email không hợp lệ
Mật khẩu	- Bắt buộc (required)	- Vui lòng nhập mật khẩu
Đăng nhập	- Email & mật khẩu đúng với tài khoản đã đăng ký	- Email hoặc mật khẩu không đúng
Nhớ đăng nhập	- (Không bắt buộc, chỉ lưu trạng thái session)	(Không có lỗi, chỉ là lựa chọn tùy ý)

#### Chi tiết thông báo lỗi và vị trí hiển thị

- **Trường Email**
  - Nếu để trống: *"Vui lòng nhập email"*
  - Nếu sai định dạng: *"Email không hợp lệ"*
- **Trường Mật khẩu**
  - Nếu để trống: *"Vui lòng nhập mật khẩu"*
- **Trường hợp sai thông tin**

- Nếu nhập email và/hoặc mật khẩu sai:  
"Email hoặc mật khẩu không đúng", thường hiển thị chung phía dưới form hoặc gần nút Đăng nhập.

## Bổ sung về UX/UI

- **Loading:** Có icon loading/spinner khi chờ xác thực.
- **Tab chuyển giữa trường Email/Mật khẩu dễ dàng (accessible).**
- **Hiển thị thông báo lỗi sát với trường nhập bị lỗi, màu đỏ nổi bật, font nhỏ.**
- **Đăng nhập thành công:**
  - Chuyển hướng sang Dashboard/trang chủ.
  - Hoặc hiển thị “Đăng nhập thành công!” (có thể tùy chọn show toast message).

## Test case phổ biến

Trường hợp kiểm tra	Kết quả mong đợi
Để trống email, nhập mật khẩu	Báo lỗi “Vui lòng nhập email”
Để trống mật khẩu, nhập email	Báo lỗi “Vui lòng nhập mật khẩu”
Email sai định dạng	Báo lỗi “Email không hợp lệ”
Nhập đúng định dạng email, mật khẩu nhưng sai tài khoản	Báo lỗi “Email hoặc mật khẩu không đúng”
Nhập đúng, nhấn “Đăng nhập”	Đăng nhập thành công, chuyển trang/hiển thị thành công
Tích hoặc bỏ “Nhớ đăng nhập”	Chỉ ảnh hưởng tới session, không có lỗi

## Ghi chú cho developer & tester

- Không được tiết lộ thông tin nhạy cảm (VD: “Email đúng, sai mật khẩu!”).
- Không show lỗi chi tiết backend cho người dùng cuối.
- Luôn báo lỗi “Email hoặc mật khẩu không đúng” chung chung với đăng nhập thất bại.

## - Màn hình FR03 – Tạo mới Task:

Field	Validation Rule	Error Message (Thông báo lỗi)
Tên task	- Bắt buộc (required)- Không chỉ toàn khoảng trắng- Giới hạn độ dài (VD: 100 ký tự)	- Vui lòng nhập tên task- Tên task không được quá dài
Mô tả	- Không bắt buộc- Nếu có: Giới hạn độ dài (VD: 500 ký tự)	- Mô tả không được quá dài
Deadline	- Bắt buộc (required)- Phải chọn ngày hợp lệ ( $\geq$ hôm nay)- Đúng định dạng ngày	- Vui lòng chọn deadline- Deadline không hợp lệ
Ưu tiên	- Bắt buộc (required)- Chỉ chọn 1 giá trị hợp lệ (Low, Medium, High)	- Vui lòng chọn mức ưu tiên
Checklist	- Không bắt buộc- Mỗi item không vượt quá 100 ký tự	- Checklist không được để trống

## Chi tiết và UX/UI

- **Nút “Tạo mới”**
  - Chỉ enable khi nhập đủ Tên task, Deadline, Ưu tiên.
  - Nếu có trường lỗi, show lỗi phía dưới input đó, màu đỏ, font nhỏ.
- **Checklist**
  - Thêm/xoá động, báo lỗi nếu để trống checklist khi thêm mới.
- **Form luôn cho phép nhấn “Huỷ” để thoát mà không lưu.**
- **Deadline phải từ hôm nay trở đi (không cho chọn ngày quá khứ).**

## Test case phổ biến

Trường hợp kiểm tra	Kết quả mong đợi
Không nhập tên task, nhấn Tạo mới	Báo lỗi “Vui lòng nhập tên task”
Tên task > 100 ký tự	Báo lỗi “Tên task không được quá dài”
Không chọn deadline, nhấn Tạo mới	Báo lỗi “Vui lòng chọn deadline”
Chọn deadline trong quá khứ	Báo lỗi “Deadline không hợp lệ”

Không chọn ưu tiên	Báo lỗi “Vui lòng chọn mức ưu tiên”
Checklist: để trống item rồi nhấn thêm	Báo lỗi “Checklist không được để trống”
Mô tả quá dài (nếu có limit)	Báo lỗi “Mô tả không được quá dài”
Nhập đủ thông tin hợp lệ, nhấn Tạo mới	Tạo task thành công, reset form/chuyển trang

**- Màn hình FR04 & FR06 – Xem danh sách task:**

Field/Chức năng	Validation Rule / Kiểm tra	Error Message (Thông báo lỗi)
<b>Tìm kiếm tên task</b>	- Không bắt buộc- Nếu nhập, loại bỏ ký tự trắng đầu/cuối	(Không cần báo lỗi, chỉ hiển thị “Không tìm thấy task nào” nếu không có kết quả)
<b>Lọc trạng thái</b>	- Chỉ chọn 1 giá trị (Tất cả, Đang làm, Đã hoàn thành, ...)	(Không cần báo lỗi – dropdown, chỉ lọc lại list)
<b>Lọc ưu tiên</b>	- Chỉ chọn 1 giá trị hợp lệ (Tất cả, High, Medium, Low)	(Không cần báo lỗi – dropdown, chỉ lọc lại list)
<b>Lọc deadline</b>	- Chỉ chọn ngày hợp lệ (Từ ngày, Đến ngày, không được đảo ngược nếu chọn khoảng)	- Ngày bắt đầu phải nhỏ hơn hoặc bằng ngày kết thúc (nếu chọn range)- Ngày không hợp lệ
<b>Checkbox trạng thái</b>	- Luôn hợp lệ, cho phép tick/untick	(Không cần báo lỗi – cập nhật trạng thái task)
<b>Nút Sửa</b>	- Task phải tồn tại, dữ liệu phải hợp lệ	- Task không tồn tại hoặc dữ liệu lỗi (nếu API trả về lỗi khi sửa task)
<b>Phân trang</b>	- Trang số hợp lệ, không vượt quá tổng số trang	(Không cần báo lỗi – disable nút nếu out of range)

**Các trường hợp hợp lỗi/thông báo hay gặp:**

- **Kết quả tìm kiếm/truy vấn không có task:**
  - Hiển thị: “*Không tìm thấy task nào phù hợp với điều kiện lọc/tìm kiếm*”
- **Chọn deadline sai logic (Từ ngày > Đến ngày):**
  - Hiển thị: “*Ngày bắt đầu phải nhỏ hơn hoặc bằng ngày kết thúc*”
- **API lỗi khi sửa task hoặc cập nhật trạng thái:**
  - Hiển thị: “*Có lỗi xảy ra, vui lòng thử lại sau*”

**UX/UI lưu ý:**

- Nếu không có kết quả, thay danh sách bằng dòng thông báo ở giữa bảng.
- Disable phân trang ở đầu/cuối khi out of range.
- Khi thao tác nhanh (tìm kiếm, lọc), loading spinner nhỏ có thể hiển thị.
- Trạng thái task sẽ update ngay trên UI, có thể dùng optimistic update.

#### Test case phổ biến:

Trường hợp kiểm tra	Kết quả mong đợi
Tìm kiếm task với từ khoá không tồn tại	Hiển thị “Không tìm thấy task nào...”
Chọn deadline từ ngày > đến ngày	Hiển thị lỗi “Ngày bắt đầu phải nhỏ hơn...”
Chọn ưu tiên, trạng thái bất kỳ	Danh sách lọc đúng, không báo lỗi
Tick/untick hoàn thành task	Trạng thái cập nhật, không báo lỗi
Nhấn sửa task trên dòng không tồn tại	Báo lỗi hoặc reload lại list
Chuyển trang ngoài vùng	Phân trang bị disable, không cho thao tác

#### Ghi chú:

- Phần lớn validation là UI/UX, ít khi báo lỗi nếu chọn filter.
- Trường hợp API hoặc backend lỗi, luôn show thông báo chung dễ hiểu cho user.

#### - Màn hình FR05 - Cập nhật, xóa task:

Field	Validation Rule	Error Message (Thông báo lỗi)
<b>Tên task</b>	- Bắt buộc (required)- Không chỉ khoảng trắng- Giới hạn độ dài (VD: 100 ký tự)	- Vui lòng nhập tên task- Tên task không được quá dài
<b>Mô tả</b>	- Không bắt buộc- Nếu có: Giới hạn độ dài (VD: 500 ký tự)	- Mô tả không được quá dài
<b>Deadline</b>	- Bắt buộc (required)- Phải chọn ngày hợp lệ (>= hôm nay)- Đúng định dạng ngày	- Vui lòng chọn deadline- Deadline không hợp lệ



<b>Ưu tiên</b>	- Bắt buộc (required)- Chỉ chọn 1 giá trị hợp lệ (Low, Medium, High)	- Vui lòng chọn mức ưu tiên
<b>Checklist</b>	- Không bắt buộc- Nếu có: mỗi item không được trống, không vượt quá 100 ký tự- Cho phép tick/untick- Cho phép xoá từng item	- Checklist không được để trống

### Nút hành động và xác nhận

Action	Validation / Rule	Error Message (Thông báo lỗi)
<b>Cập nhật</b>	- Chỉ enable khi dữ liệu hợp lệ ở các trường bắt buộc	Hiển thị lỗi tương ứng từng trường nếu sai
<b>Xoá task</b>	- Có popup xác nhận ("Bạn có chắc chắn muốn xoá?")	(Không cần thông báo lỗi nếu user huỷ thao tác)
<b>Huỷ</b>	- Không validate, quay lại màn trước	(Không có lỗi)

### Trường hợp lỗi/thông báo thường gặp

- **Để trống tên task:**
  - “Vui lòng nhập tên task”
- **Tên task quá dài:**
  - “Tên task không được quá dài”
- **Không chọn deadline:**
  - “Vui lòng chọn deadline”
- **Deadline nhỏ hơn hôm nay:**
  - “Deadline không hợp lệ”
- **Không chọn ưu tiên:**
  - “Vui lòng chọn mức ưu tiên”
- **Checklist để trống khi thêm mới:**
  - “Checklist không được để trống”
- **API lỗi khi cập nhật/xoá:**

- “Có lỗi xảy ra khi cập nhật/xoá task, vui lòng thử lại sau”

- **Khi xoá task:**

- Popup xác nhận: “Bạn có chắc chắn muốn xoá task này không? Hành động này không thể hoàn tác.”

## UX/UI lưu ý

- Luôn xác nhận khi xoá (modal/popup).
- Lỗi hiển thị sát trường nhập, màu đỏ, font nhỏ.
- Nút "Cập nhật" chỉ enable khi hợp lệ, "Xoá" luôn enable (nhưng luôn xác nhận trước khi xoá).
- Khi thao tác thành công, có thể show toast/thông báo (“Cập nhật thành công!”, “Xoá task thành công!”).

## Test case phổ biến

Trường hợp kiểm tra	Kết quả mong đợi
Để trống tên task, nhấn cập nhật	Báo lỗi “Vui lòng nhập tên task”
Chọn deadline quá khứ	Báo lỗi “Deadline không hợp lệ”
Sửa checklist, để trống rồi lưu	Báo lỗi “Checklist không được để trống”
Nhấn Xoá task, xác nhận	Task bị xoá, thông báo thành công, quay lại
Nhấn Xoá task, huỷ xác nhận	Không bị xoá, đóng popup xác nhận
Nhập đủ thông tin, nhấn Cập nhật	Task được cập nhật, hiển thị thông báo thành công

-