مرجع التعامل مع IndexedDB في مشروع TaskMaster

# فهرس المحتويات

1. 1. تنصيب مكتبة idb
2. 2. إنشاء قاعدة البيانات
3. 3. فتح قاعدة البيانات
4. 4. إضافة مهمة إلى IndexedDB
5. 5. قراءة كل المهام
6. 6. حذف مهمة واحدة
7. 7. حذف جميع المهام
8. 8. تحويل صورة إلى Base64
9. 9. تخزين صورة داخل IndexedDB
10. 10. مزامنة المهام مع Supabase
11. 11. رفع الصور إلى Supabase Storage
12. 12. عرض المهام الأوفلاين في الواجهة
13. 13. حذف مهمة أوفلاين من الواجهة

# 1. تنصيب مكتبة idb

قم بتنصيب مكتبة idb عبر npm:  
```  
npm install idb  
```

# 2. إنشاء قاعدة البيانات

نقوم بإنشاء قاعدة بيانات باسم TaskMasterDB تحتوي على Object Store باسم tasks:  
```js  
import { openDB } from 'idb';  
  
export const openDB = () => {  
 return idb.openDB('TaskMasterDB', 1, {  
 upgrade(db) {  
 if (!db.objectStoreNames.contains('tasks')) {  
 db.createObjectStore('tasks', {  
 keyPath: 'id',  
 autoIncrement: true  
 });  
 }  
 }  
 });  
};  
```

# 3. فتح قاعدة البيانات

تتم عملية الفتح عند كل استخدام للـ DB:  
```js  
const db = await openDB();  
```

# 4. إضافة مهمة إلى IndexedDB

```js  
export const addTaskToDB = async (task) => {  
 const db = await openDB();  
 const tx = db.transaction("tasks", "readwrite");  
 const store = tx.objectStore("tasks");  
  
 const taskToStore = { ...task };  
  
 return new Promise((resolve, reject) => {  
 if (task.image instanceof File) {  
 const reader = new FileReader();  
 reader.onload = () => {  
 taskToStore.imageBlob = reader.result;  
 delete taskToStore.image;  
 store.add(taskToStore);  
 resolve();  
 };  
 reader.onerror = () => reject(reader.error);  
 reader.readAsDataURL(task.image);  
 } else {  
 store.add(taskToStore);  
 resolve();  
 }  
 });  
};  
```

# 5. قراءة كل المهام

```js  
export const getAllTasksFromDB = async () => {  
 const db = await openDB();  
 return db.getAll("tasks");  
};  
```

# 6. حذف مهمة واحدة

```js  
export const deleteTaskFromDB = async (id) => {  
 const db = await openDB();  
 return db.delete("tasks", id);  
};  
```

# 7. حذف جميع المهام

```js  
export const clearAllTasksFromDB = async () => {  
 const db = await openDB();  
 return db.clear("tasks");  
};  
```

# 8. تحويل صورة إلى Base64

يتم تلقائياً عند استخدام FileReader كما في دالة `addTaskToDB`.

# 9. تخزين صورة داخل IndexedDB

يتم تخزين الصورة داخل `imageBlob` كـ base64 داخل الـ IndexedDB.

# 10. مزامنة المهام مع Supabase

```js  
export const syncOfflineTasks = async (dispatch) => {  
 const userId = getOrCreateUserId();  
 const offlineTasks = await getAllTasksFromDB();  
  
 for (const task of offlineTasks) {  
 const { name, description, priority, number, classification, imageBlob } = task;  
  
 let imageUrl = null;  
  
 if (imageBlob) {  
 try {  
 imageUrl = await uploadBase64ImageToSupabase(imageBlob, userId);  
 } catch (err) {  
 console.error("فشل رفع الصورة:", err.message);  
 }  
 }  
  
 const { data, error } = await supabase.from("tasks").insert([{  
 taskName: name,  
 taskDescription: description,  
 numberSelection: number,  
 prioritySelection: priority,  
 classification,  
 completed: false,  
 imageUrl,  
 userId,  
 }]).select();  
  
 if (!error) {  
 dispatch(addNewTask(data[0]));  
 await deleteTaskFromDB(task.id);  
 } else {  
 console.error("فشل رفع المهمة:", error.message);  
 }  
 }  
};  
```

# 11. رفع الصور إلى Supabase Storage

```js  
export async function uploadBase64ImageToSupabase(base64String, userId) {  
 const fileName = `tasks/${userId}/${Date.now()}.png`;  
  
 const { data, error } = await supabase.storage  
 .from("pictures")  
 .upload(fileName, base64ToBlob(base64String), {  
 contentType: "image/png",  
 });  
  
 if (error) throw new Error(error.message);  
  
 const { data: publicUrlData, error: publicUrlError } = supabase.storage  
 .from("pictures")  
 .getPublicUrl(fileName);  
  
 if (publicUrlError) throw new Error(publicUrlError.message);  
  
 return publicUrlData.publicUrl;  
}  
```

# 12. عرض المهام الأوفلاين في الواجهة

تم استخدام `setLocalTasks` و `localTasks.map()` لعرض المهام، ويتم استخدام `task.imageBlob` داخل عنصر `<img src={task.imageBlob} />`.

# 13. حذف مهمة أوفلاين من الواجهة

```js  
const handleDelete = async (id) => {  
 await deleteTaskFromDB(id);  
 const updated = await getAllTasksFromDB();  
 setLocalTasks(updated);  
};  
```