Project 1 Reflection

Team_2

Team members:

Yucheng An (989444449)

Henglay Eung (989468833)

Harsh Umesh Bhanushali (989467927)

Task 1:

Define an **IndexedDB** object store named "TodoList" and populate it with 100,000 randomly generated objects, each containing `task`, `status`, and `dueDate` properties and display it on the console or the browser.

We learned to populate large amounts of data (100k random objects) into IndexedDB with efficient use of transactions. Start from building DB, limit data format, data generated functions, to add objects into IndexedDB. We need to deal with how to prevent duplicated generating the same data when the page refreshes.

Task 1:

Set 1000 objects to status "completed" and the remaining ones to status "progress"

We learned updating statuses for 1000 tasks to "completed" and the rest to "pending" was successful. We faced some challenges on managing and keeping counters synchronized during cursor traversal required attention to avoid incorrect status assignments.

Task 2:

Measure and display the time (in milliseconds) required to read all objects with `status` set to "completed" on the console or the browser

We learned how to measure and display the time between one action to another action. Because of async and await functions, we cannot deal with measures in the normal way. We used function parameters to transfer the action we want to do such as assigning status, what status we want to set, decide different flags.

Task 3:

Apply a **read-only flag** to the object store and measure and display the time to read all completed tasks again on the console or the browser.

We learned to switch different modes between “readwrite” and “read-only” to access the data and re-set like task 3. To make code clear and readable, we design one function to pass parameters as different statuses. Read-only mode improves efficiency over “readwrite” mode in IndexedDB, especially in the field of large data processing.

Task 4:

Create an index on the `status` field, then measure and display the time to read all completed tasks on the console or the browser.

We learned we can create different index fields by using different parameters such as id, status, date, etc. This will significantly improve the efficiency for reading all status with ‘completed’. We need to know before, the index must be created during the “onupgradeneeded” process of IndexedDB initialization. Without this, the index might not exist, leading to errors while running the code.

Task 5:

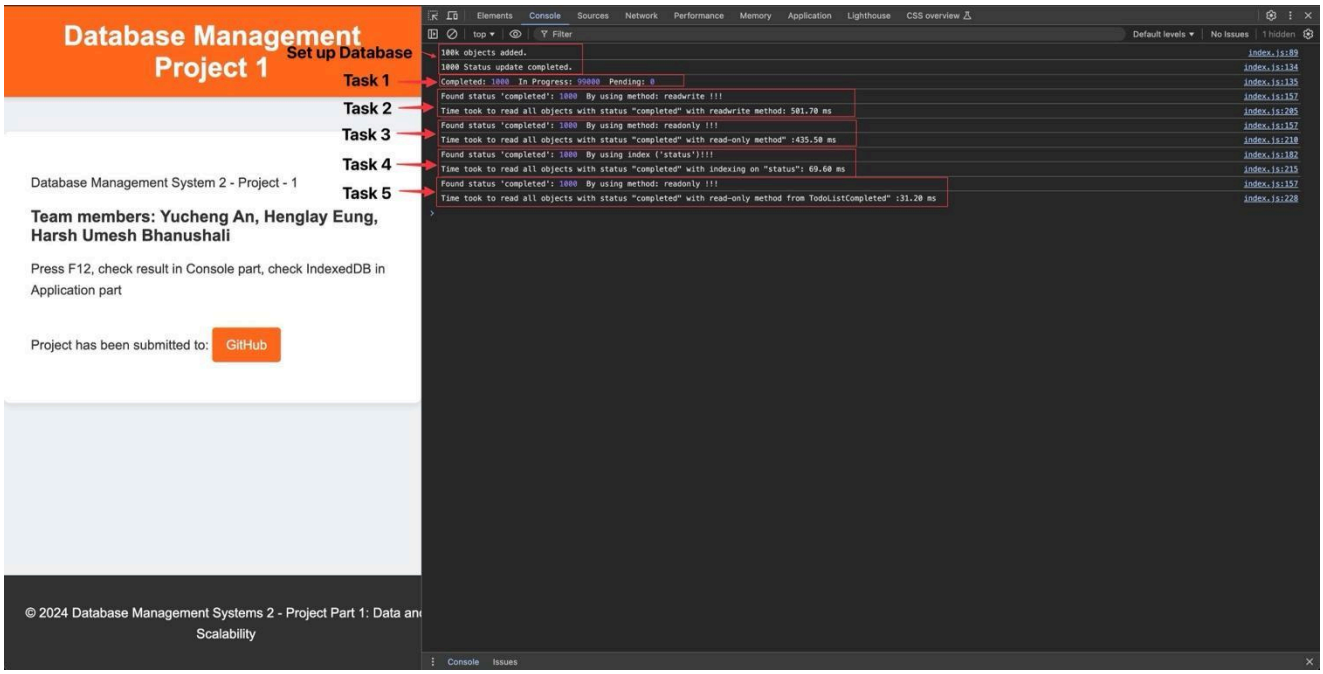
Define a new object store called "TodoListCompleted", copy all completed tasks from "TodoList" to this new store, and measure and display the time required to read all completed tasks from "TodoListCompleted" on the console or the browser.

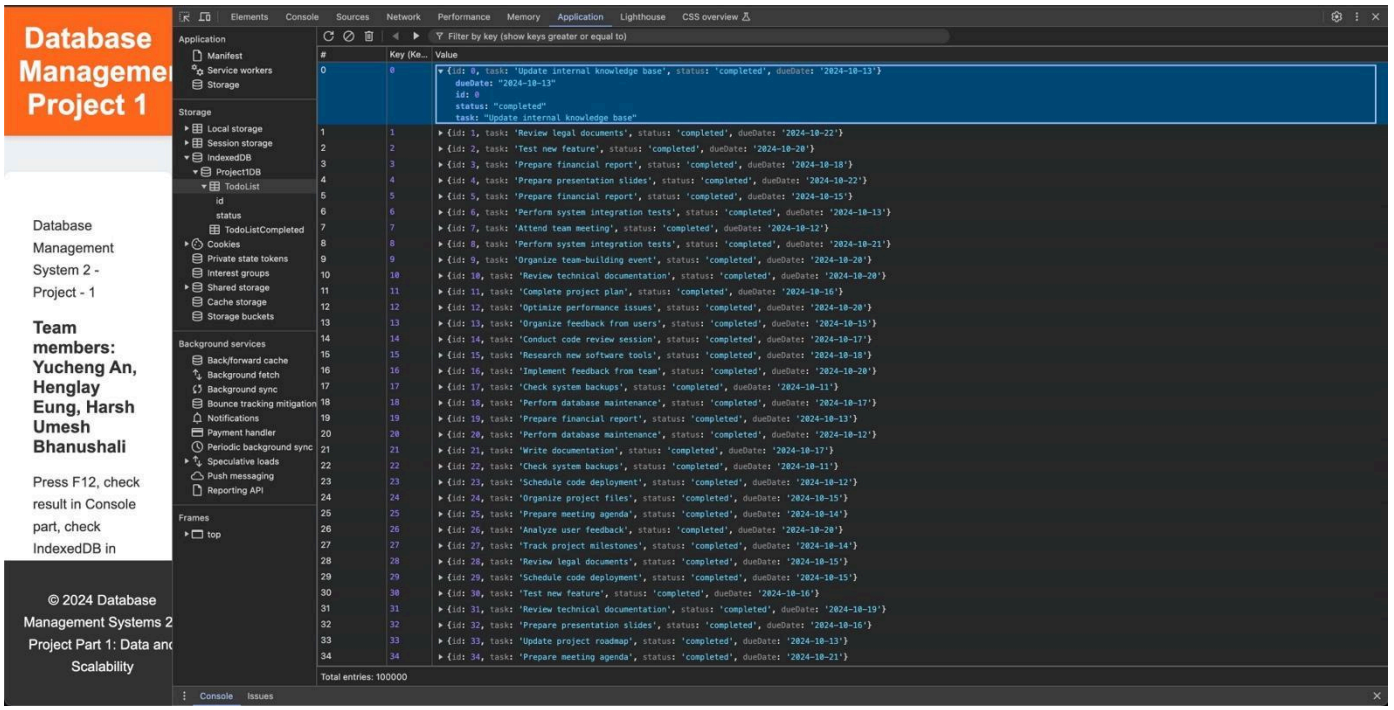
We learned that we could use it to create a new store for some special cast in this example, we make a new one to save that task status are “completed”. We may separate the object store for different states, dates, or names. This will make it more efficient and easier to structure of data while we are dealing with large datasets.

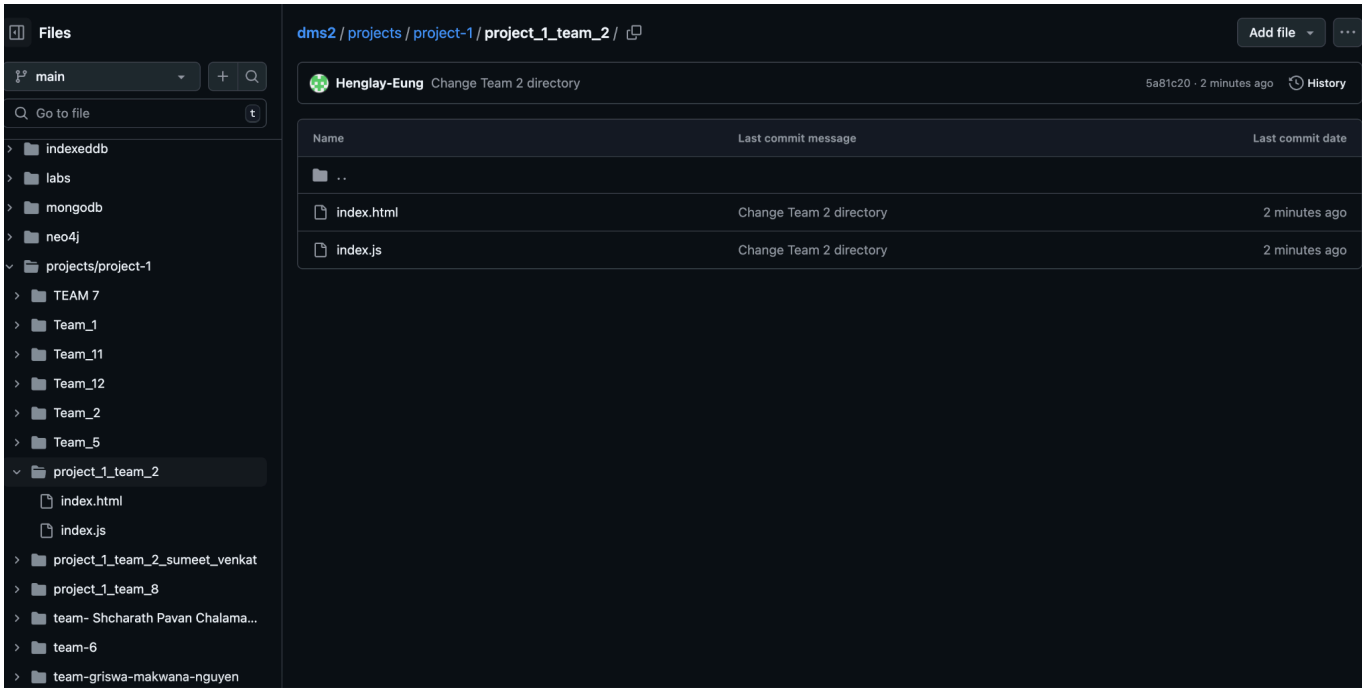
Key Insights

After finishing this project and previous labs related to IndexedDB, IndexedDB is efficient for large datasets based on the local browser but requires careful transaction management, and index setting, especially with cursor traversal. Correctly using the correct different transaction modes such as read-only, and readwrite in special casts can significantly improve performance. (We already compare by using this project). Proper indexing is essential for optimizing a lot of performance, and creating an index on the right field is crucial. Not only that, we can also re-organize data into separate object stores based on states or categories (like completed tasks, a special range of date, or more variables) can simplify querying and improve maintainability for large databases.

Screenshot


Database Management Project 1
Set up Database
Task 1
Task 2
Task 3
Task 4
Task 5
Database Management System 2 - Project - 1
Team members: Yucheng An, Henglay Eung, Harsh Umesh Bhanushali
Press F12, check result in Console part, check IndexedDB in Application part
Project has been submitted to:
GitHub
© 2024 Database Management Systems 2 - Project Part 1: Data and Scalability
Elements Console Sources Network Performance Memory Application Lighthouse CSS overview
100k objects added.
1000 Status update completed.
Completed: 1000 In Progress: 99000 Pending: 0
Found status 'completed': 1000 By using method: readwrite !!!
Time took to read all objects with status "completed" with readwrite method: 501.70 ms
Found status 'completed': 1000 By using method: readonly !!!
Time took to read all objects with status "completed" with read-only method" :435.50 ms
Found status 'completed': 1000 By using index ('status')!!!
Time took to read all objects with status "completed" with indexing on "status": 69.60 ms
Found status 'completed': 1000 By using method: readonly !!!
Time took to read all objects with status "completed" with read-only method from TodoListCompleted" :31.20 ms
Console Issues


Database Management Project 1
Database Management System 2 - Project - 1
Team members: Yucheng An, Henglay Eung, Harsh Umesh Bhanushali
Press F12, check result in Console part, check IndexedDB in
© 2024 Database Management Systems 2 Project Part 1: Data and Scalability
Elements Console Sources Network Performance Memory Application Lighthouse CSS overview
Application
Manifest
Service workers
Storage
Storage
Local storage
Session storage
IndexedDB
Project1DB
TodoList
id
status
TodoListCompleted
Cookies
Private state tokens
Interest groups
Shared storage
Cache storage
Storage buckets
Background services
Back/forward cache
Background fetch
Background sync
Bounce tracking mitigation
Notifications
Payment handler
Periodic background sync
Speculative loads
Push messaging
Reporting API
Frames
top
Filter by key (show keys greater or equal to)
Key (Ke...
Value
Total entries: 100000
Console Issues


Files
main
Go to file
indexeddb
labs
mongodb
neo4j
projects/project-1
TEAM 7
Team_1
Team_11
Team_12
Team_2
Team_5
project_1_team_2
index.html
index.js
project_1_team_2_sumeet_venkat
project_1_team_8
team- Shcharath Pavan Chalama...
team-6
team-griswa-makwana-nguyen
dms2 / projects / project-1 / project_1_team_2 /
Add file
Henglay-Eung Change Team 2 directory
5a81c20 · 2 minutes ago
History
Name
Last commit message
Last commit date
..
index.html
Change Team 2 directory
2 minutes ago
index.js
Change Team 2 directory
2 minutes ago