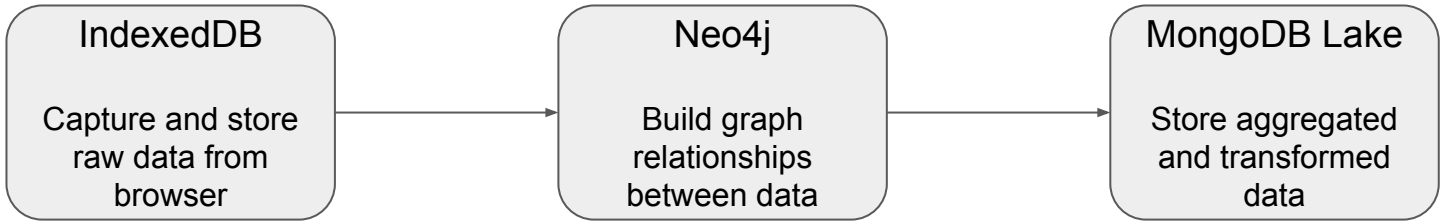
Lab3

Yu-Tai Lee - 989505348

Architecture & Data Flow

- IndexedDB: Used to capture and store raw data directly from the browser.
- Neo4j: Builds graph relationships between users, entities, or transactions.
- MongoDB Lake: Acts as the centralized repository for aggregated and cleaned data.
- Process:
  - Extract data from IndexedDB
  - Transform and link through Neo4j relationships
  - Load final structured results into MongoDB for analytics



Lessons Learned

- Learned how to connect heterogeneous databases using a shared JSON schema.
- The graph model in Neo4j helps visualize data relationships that were hidden in raw data.
- Handling asynchronous data flow between client and server taught us about synchronization issues.
- Gained a stronger understanding of data validation and error handling during transfers.

Challenges

- Main Challenges:
  - Difficult to maintain data consistency between Neo4j and MongoDB due to schema flexibility.
  - Migration speed was slower when the dataset grew larger.
  - Debugging cross-database connections took time because of different query languages.
- What Worked:
  - Visualizing data in Neo4j was intuitive and made debugging easier.
  - MongoDB handled final storage efficiently once the structure was cleaned.
- Future Improvements:
  - Implement automated data synchronization scripts.
  - Add real-time monitoring dashboards to track data flow status.