

UI/UX Design Document for Key-Value Store Manager

Project Overview

The **Key-Value Store Manager** is a web-based tool implemented using Streamlit to manage a key-value database stored in a JSON Lines (output.jsonl) file. The UI allows users to input commands, process them, and view the resulting state of the database in a tabular format.

Goals

- Provide a simple, interactive interface for managing a key-value database.
- Ensure user actions like "Insert", "Update", "Delete", and "View" are intuitive and quick to execute.
- Display the current state of the database dynamically in a clean, readable format.
- Offer real-time feedback for each user action.

User Personas

Persona	Description	Goals
Database Manager	A user managing key-value data in JSONL.	Easily insert, update, delete, and view database records.
Developer	A technical user testing database CRUD.	Validate changes and get real-time feedback.

Features

Input Command Section

- **Purpose:** Accept user commands for CRUD operations.
- **Location:** Top of the page.
- **UI Components:**
 - **Label:** Enter a command:
 - **Text Input Box:**
 - Placeholder: e.g., Insert key 'user1' with value 'John Doe'
 - Function: Accept commands and trigger backend operations.
 - Action: Automatically processes the command when the user presses **Enter**.

Feedback Area

- **Purpose:** Display results of the executed command.
- **Location:** Below the input command section.
- **UI Components:**
 - **Label:** Feedback:
 - **Text Area:**

- Read-only.
- Displays success or error messages after processing the command.

Database State

- **Purpose:** Display the current state of the database in tabular format.
- **Location:** Below the Feedback Area.
- **UI Components:**
 - **Label:** Database State:
 - **Data Table:**
 - Interactive table showing:
 - Key
 - Value
 - Created Datetime
 - Updated Datetime
 - **Features:**
 - Horizontal and vertical scrolling for large datasets.
 - Dynamic updates after each command.

Information Architecture

Page Layout

Section	Content Description
Header	Title: Key-Value Store Manager
Command Input	Textbox for users to input natural language commands.
Feedback Area	Displays the output of the executed command.
Database State	Interactive table showing the database state.

Wireframe

AI agent controlling backend system

Key-Value Backend System

Enter a command

Insert the natural language command

Feedback

Database State

Logs By Hour

Interaction Design

Command Input Box

- **Action:** Processes commands on **Enter**.
 - Validation:
 - Empty input: Show error: Please enter a valid command.
 - Invalid commands: Show error: Unknown command.
- **Feedback Area**
 - Displays:
 - Success messages (e.g., Inserted key 'user1' with value 'John Doe').
 - Error messages (e.g., Key 'user2' not found.).
 - Real-time updates.

Database Table

- **Dynamic Updates:**
 - Reflects changes immediately after each command.
- **Interactivity:**
 - Scrolling for large datasets.
 - Table rows update automatically after insert/update/delete operations.
 - Sorting for each column

Logs by hour

- **Bar Graph:**
 - Visualizes logs per hour in form of bar graph.

User Flow

1. **Start:**
 - User launches the app and sees an empty database.
2. **Command Input:**
 - User enters a command in the textbox.
 - App processes the command and updates the database.
3. **Feedback:**
 - App displays feedback in the text area.
4. **Database Table:**
 - User views the updated database in the table.
5. **Bar Graph:**
 - Users can visualize the logs through bar graph

Future Enhancements

- **Advanced Filters:**
 - Allow users to filter or search records in the database table.
- **Command Suggestions:**
 - Provide auto complete or suggestions for valid commands.