DSA PROJECT

Autocomplete Feature

Using Trie Data Structure

DSA PROJECT

Introduction to the Project

What is Autocomplete?

- Autocomplete predicts words as users type, enhancing user experience in search engines, text editors, and messaging apps.

Why Autocomplete?

- It saves time, improves typing accuracy, and suggests meaningful word completions.

Problem Statement

Objective:-

Develop an efficient autocomplete system that:

- Finds words starting, ending, or containing a user-entered substring.

Challenges:-

- Managing large datasets and providing quick suggestions.
- Implementing a structure capable of searching by prefix, suffix, and infix.

Data Structure Choice - Trie

Trie (Prefix Tree):

- Efficient for storing and retrieving strings based on prefixes.
- Reduces search time complexity by storing common prefixes only once.

Suffix Trie Integration:

Added to enable suffix-based autocomplete, enhancing flexibility for users.

Features of the Autocomplete System

(BACKEND)

- Prefix Search: Predicts words that start with the user-typed prefix.
- Suffix Search: Finds words that end with specific letters.
- Infix Search: Locates words containing a particular substring.
- Word Frequency Tracking: Prioritizes frequently-used words in suggestions.

Features of the Autocomplete System (FRONTEND)

Web-Based Interface

User-friendly HTML/CSS/JavaScript frontend for easy interaction.

Socket Server on Port 8080

Listens for HTTP requests from clients, supporting both GET and POST methods.

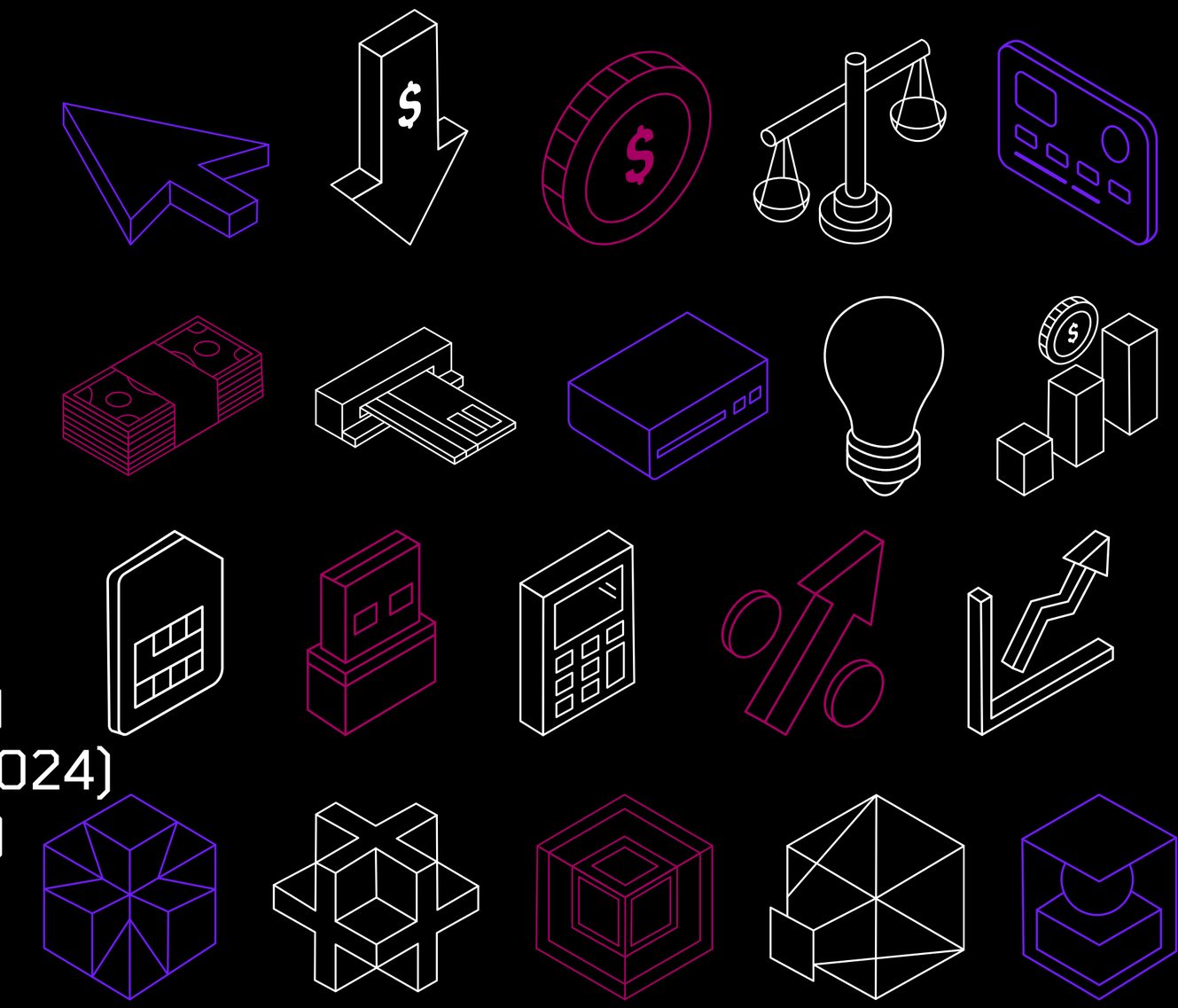
API Endpoints

POST /add-word: Adds new words to the dictionary file and Trie.

API Endpoints

GET /suggestions: Returns a JSON list of autocomplete suggestions based on the user's quer

THANK YOU



TEAM MEMBERS:-

1.JAPNEET SINGH(B23CS1022)

2.KANISHK SINGH DAYMA(B23CS1024)

3.DEVANSH SAINI(B23CS1013)