**Akshita**

**500105930**

**Batch-9**

# **Software Requirements Specification (SRS)**

**Table of content:**

Introduction

Overall description

Specific requirements

User interfaces

System interfaces

References

## **Online English Dictionary**

### **1. Introduction**

#### **1.1 Purpose**

The purpose of this document is to define the requirements for the development of an online English dictionary. This web-based application will provide users with the ability to search for word definitions,and usage examples.

#### **1.2 Scope**

The online English dictionary will be accessible via web browsers on desktop and mobile devices. It will feature a user-friendly interface for searching and displaying word information.

#### **1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* **API**: Application Programming Interface
* **UI**: User Interface
* **UX**: User Experience

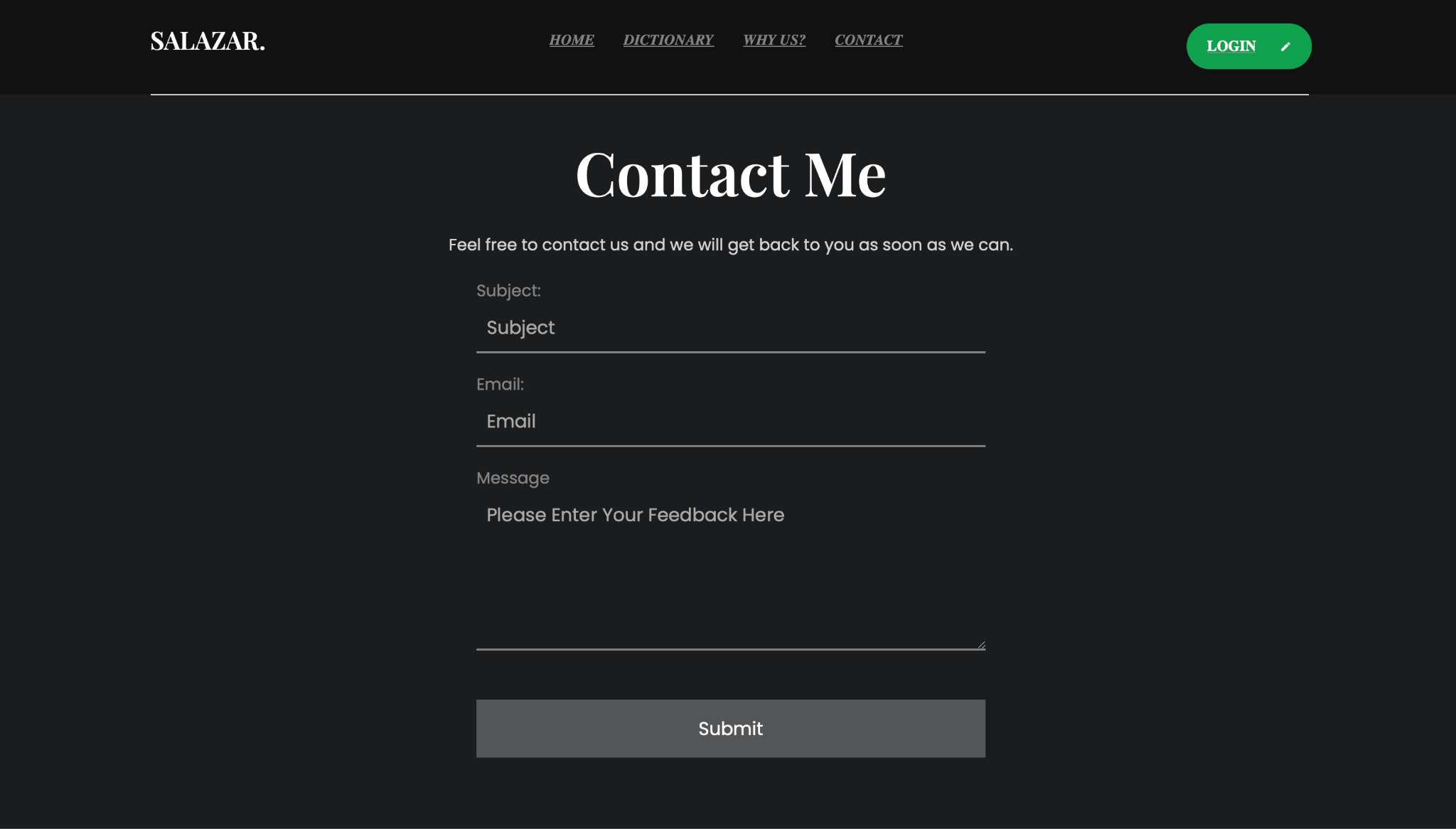
### **2. Overall Description**

#### **2.1 Product Perspective**

The online English dictionary will operate as a standalone web application. It will utilize APIs to retrieve word data from external sources and present it to users via a web-based UI.



#### **2.2 Product Functions**

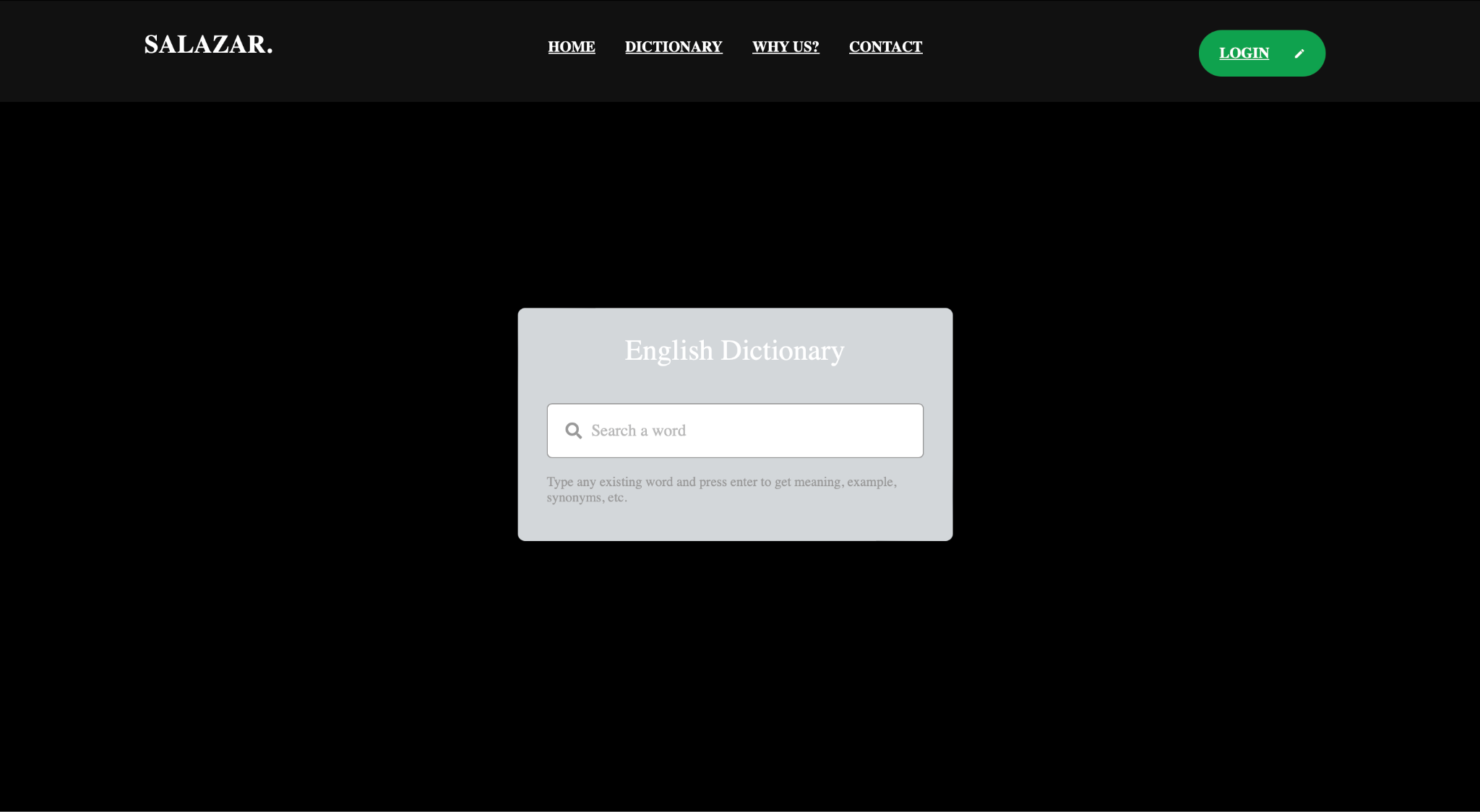
* **Word Search**: Users can search for words by typing them into a search bar.
* **Definition Display**: Display the definition of the searched word, including part of speech and pronunciation.
* **Examples**: Include usage examples of the searched word in sentences.
* **Accessibility**: Ensure the application is accessible to users with disabilities.
* 

#### **2.3 User Characteristics**

The target users of the online English dictionary include students, professionals, and individuals seeking to improve their English language skills. Users should have basic familiarity with web browsing and searching.

### **3. Specific Requirements**

#### **3.1 Functional Requirements**

* **Search Feature**: Users can enter words in a search bar to find definitions.
* 
* **Display**: Display definitions, synonyms, antonyms, and examples in a readable format.
* **Accessibility**: Ensure compliance with web accessibility standards (e.g., WCAG).

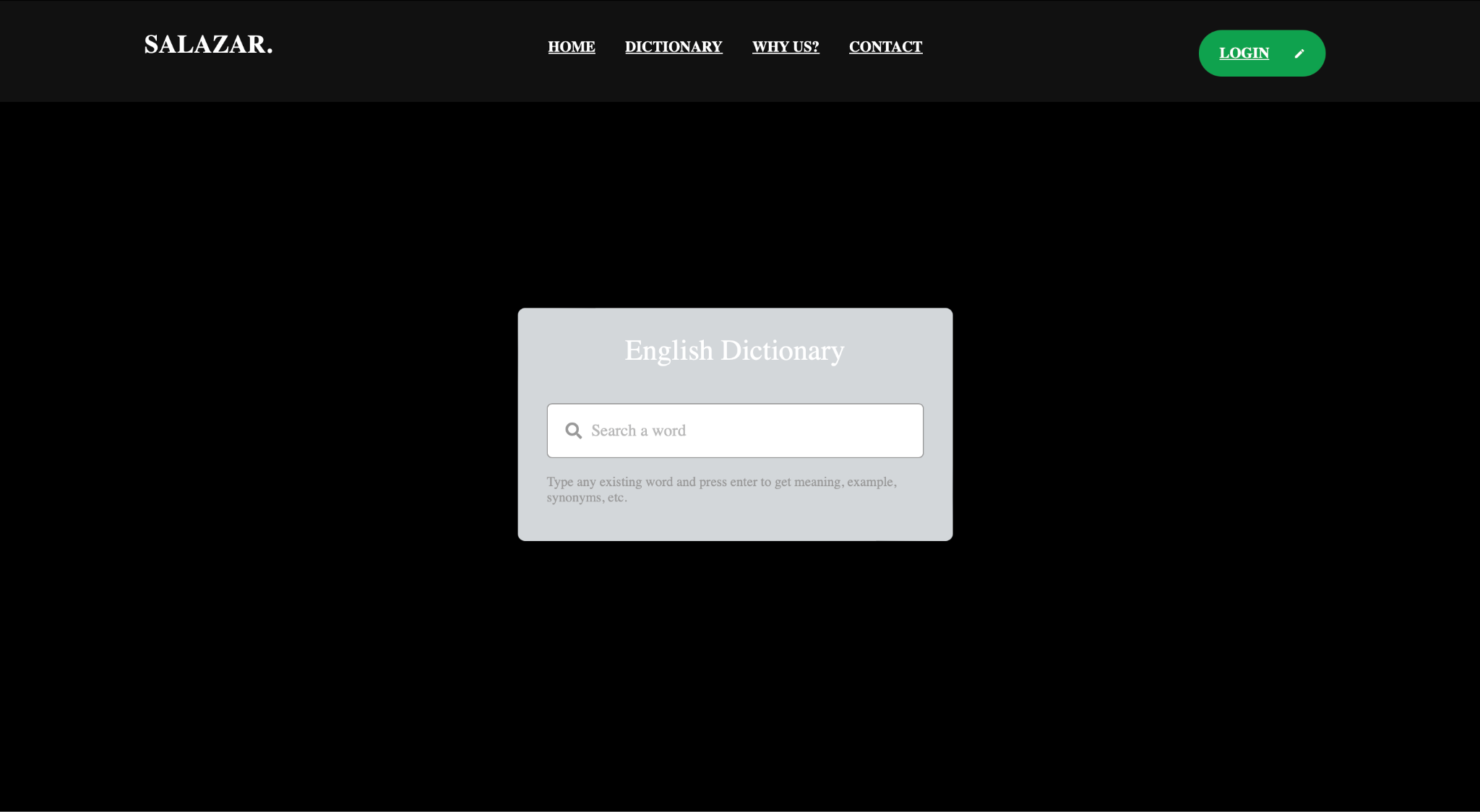
#### 

#### **3.2 Non-functional Requirements**

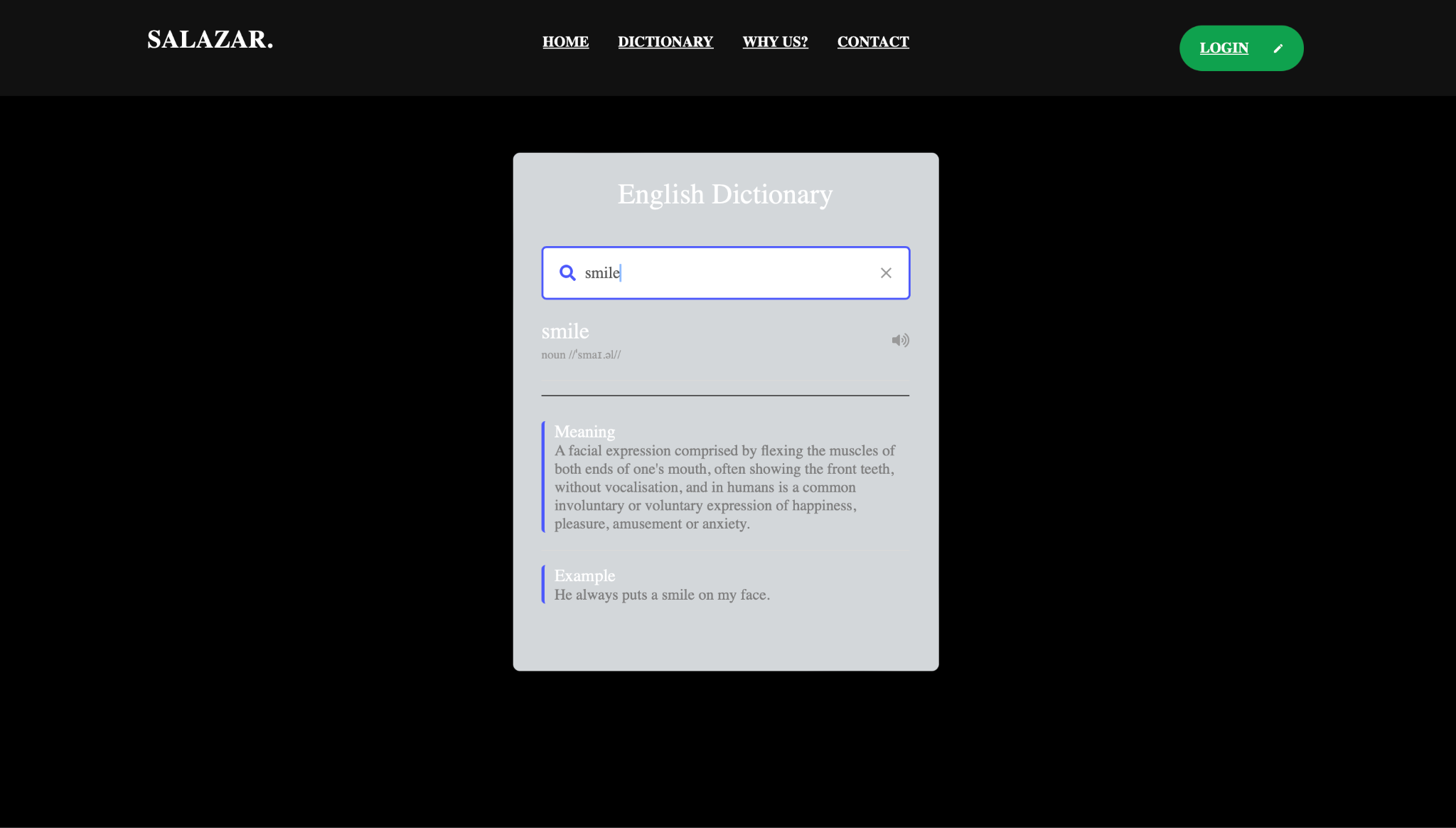
* **Performance**: The application should respond to user queries within seconds.
* **Reliability**: Ensure the application functions without errors or crashes under normal usage conditions.
* **Scalability**: The application should handle a large number of concurrent users without significant performance degradation.
* **Security**: Protect user data and prevent unauthorized access.
* **Usability**: The UI should be intuitive and easy to navigate, providing a seamless user experience.
* **Compatibility**: Ensure compatibility with major web browsers (e.g., Chrome, Firefox, Safari) and mobile devices.

### **4. User Interfaces**

#### **4.1 Search Bar**

* Provide a prominent search bar for entering words.
* 

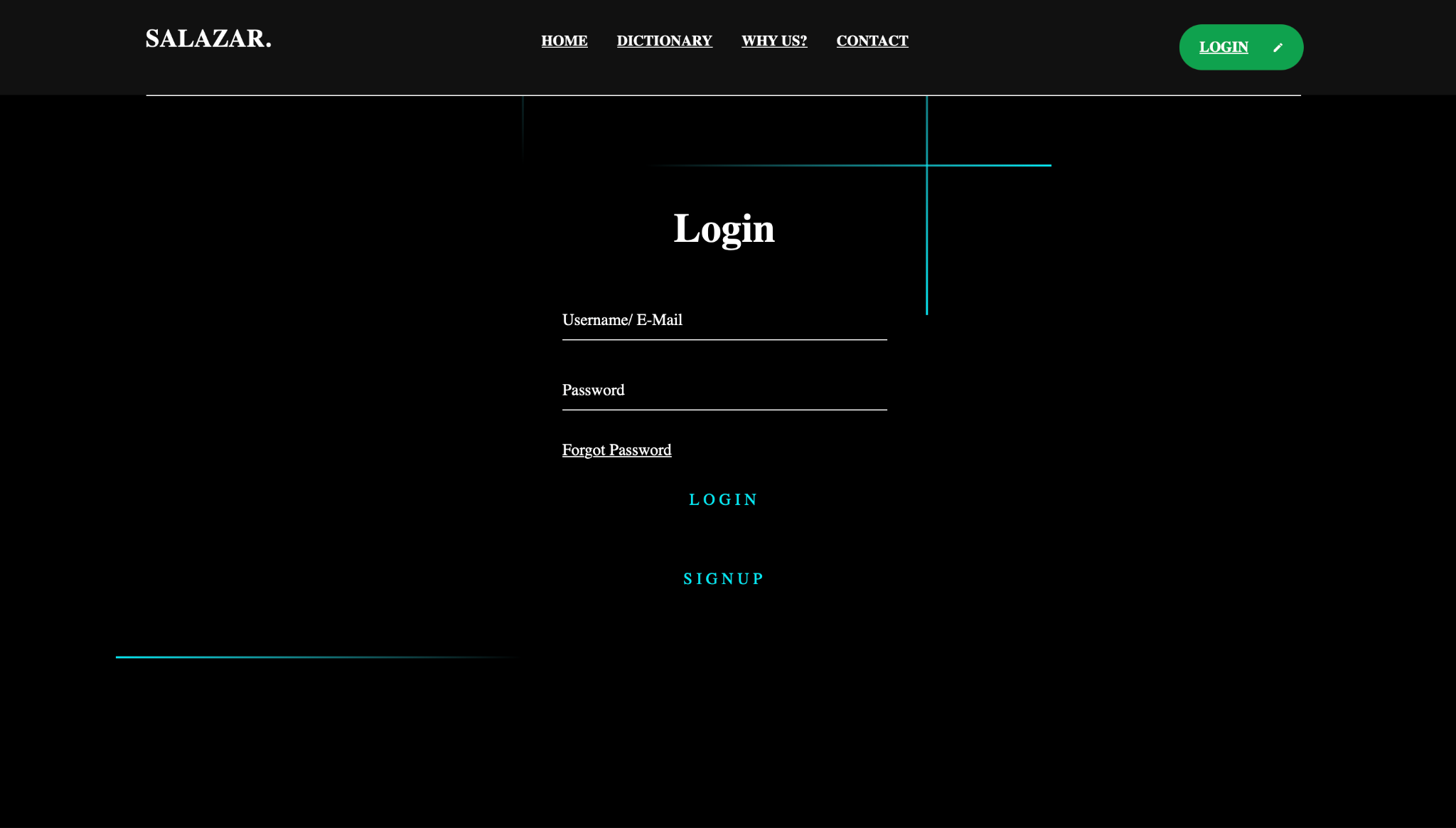
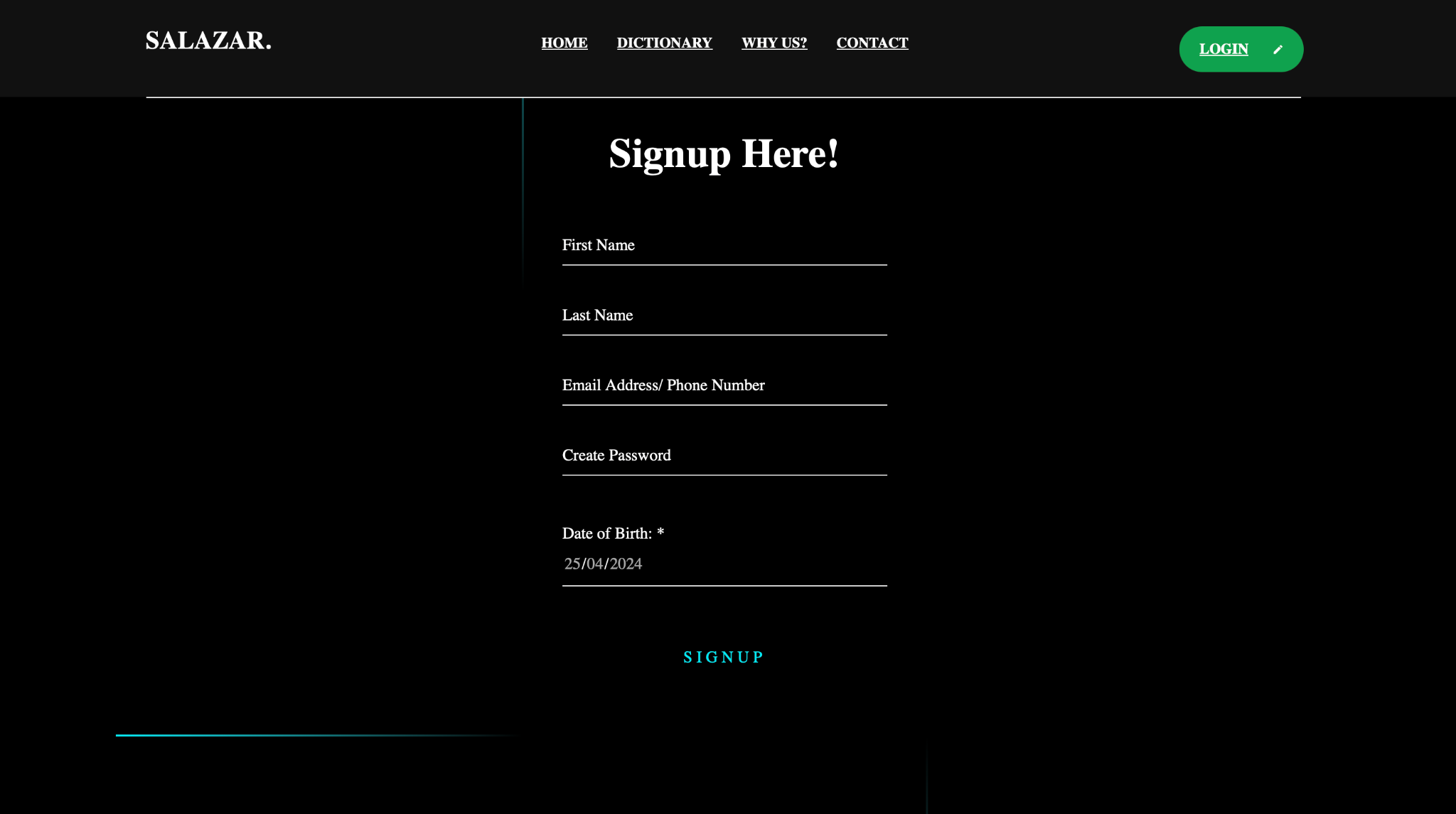
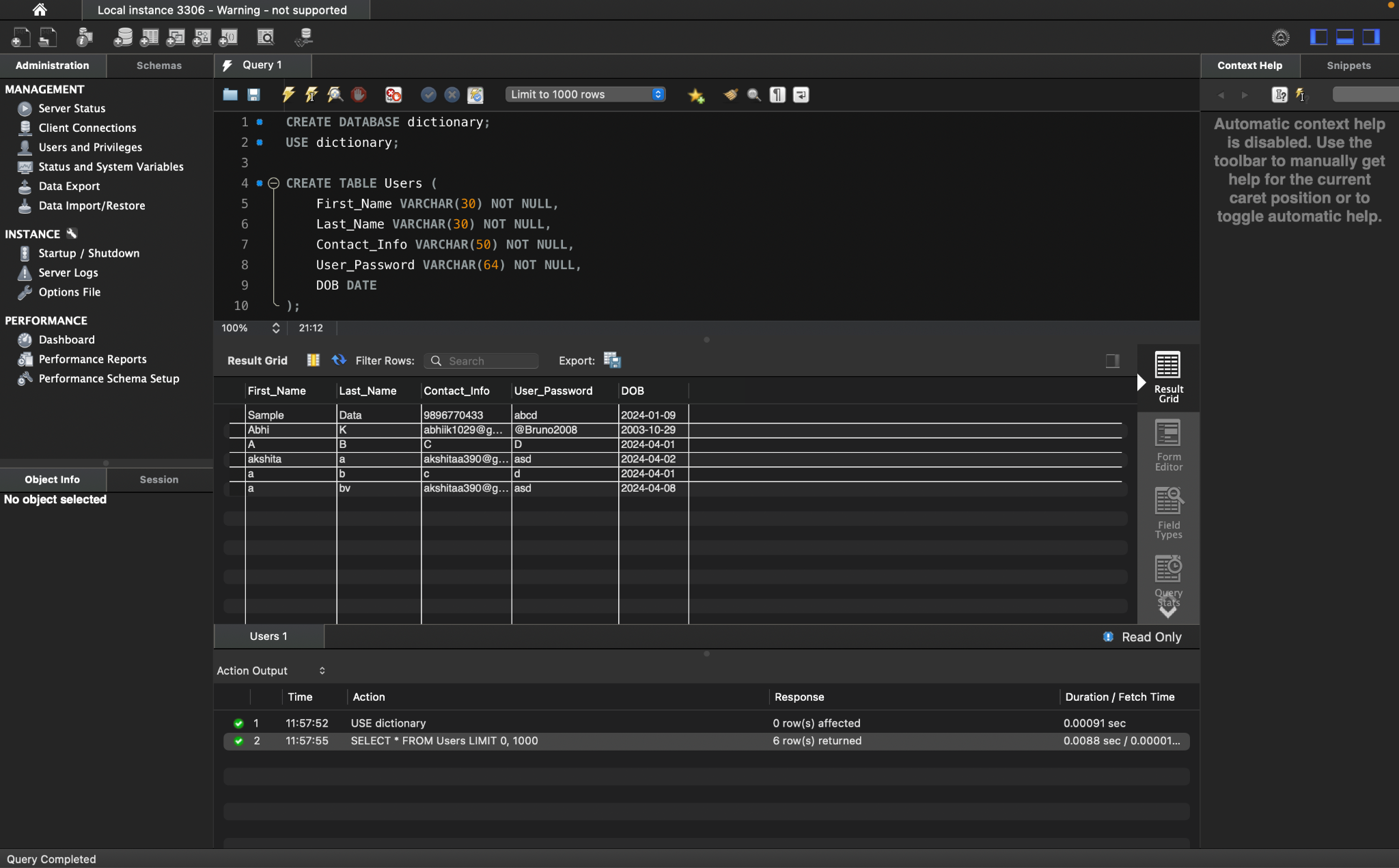
#### **4.2 Definition Display**

* Display word definitions, synonyms, antonyms, and examples in a structured layout.
* 

#### **4.3 Navigation**

* Include intuitive navigation elements (e.g., menus, buttons) for easy exploration.

### **5. System Interfaces**

* Utilize external APIs for retrieving word data (definitions, synonyms, antonyms).
* Implement a database to store user bookmarks and search history.
* 
* 
* 

**6**.  **References**

Code with harry–youtube channel for tutorials

Chatgpt and bard ai