End-Term Project Report

Mobile Application Development



Faculty name: Ms. Aarti Sangwan Student name: Ritesh (21csu392)

Piha Gupta (21csu473)

Ritik Pawar (21csu420)

Ayushi (21csu410)

Deepak (21csu408)

Semester: 6th

Group: FS-B

**Department of Computer Science and Engineering**

**The NorthCap University, Gurugram- 122001, India**

**Session 2023-24**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No** |  | **Page No.** |
| **1.** | **Project Description** | **2** |
| **2.** | **Problem Statement** | **2** |
| **3.** | **Analysis**  **3.1 Hardware Requirements**  **3.2 Software Requirements** | **2** |
| **4.** | **Design**  **4.1 Data/Input Output Description:**  **4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps** | **3** |
| **5.** | **Implementation and Testing (stage/module wise)** | **4** |
| **6.** | **Output (Screenshots)** | **5** |
| **7.** | **Conclusion and Future Scope** | **8** |

1. **Project Description**

The objective of this project is to create a language learning app called LearnIt in Flutter with modules for vocabulary, grammar, and pronunciation practice. The application Include flashcards, quizzes, and gamification elements to boost user engagement. Additionally, the app integrates with inventory management APIs for finding synonyms, antonyms, definition of the word provided by the user and uses state management for real-time updates.

1. **Problem Statement**

Language learning has become increasingly important in a globalized world. Traditional methods often lack engagement and flexibility, which can hinder progress. To address this, we aim to create an innovative language learning app called LearnIt, leveraging modern technologies and interactive methods to enhance the learning experience.

LearnIt (Education): Create a language learning app in Flutter with modules for vocabulary, grammar, and pronunciation practice. Integrate audio resources or third-party language learning APIs (if applicable).Include flashcards, quizzes, and gamification elements to boost user engagement. Employ Provider or Bloc for state management in handling user progress and scores.

**3. Analysis**

**3.1 Hardware Requirements**

* Processor: Intel Core i5 or higher
* RAM: 8 GB or higher
* Storage: 50 GB of free space
* Smartphone: Android 6.0 or higher / iOS 11.0 or higher
* Internet Connection: Stable broadband connection

**3.2 Software Requirements**

* Operating System: Windows 10 or higher / macOS / Linux
* Flutter SDK
* Dart Programming Language
* Integrated Development Environment (IDE): Android Studio / Visual Studio Code
* Firebase (for backend services)
* API Services: RESTful APIs for inventory management
* Version Control: Git

**4. Design**

**4.1 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps**

#### **Steps**

#### Vocabulary Screen

The **VocabularyScreen** displays flashcards for vocabulary learning. The screen shows a word along with its definition, synonyms, antonyms, and related words, fetched using the Datamuse API. The design includes:

* A title bar with a consistent color theme.
* Card-like designs for displaying word information.
* An interactive button to fetch the next word, updating the content dynamically.

#### Vocabulary Quiz Screen

The **VocabularyQuizScreen** offers a quiz format to test users' vocabulary knowledge. Each question presents multiple-choice answers, with the design featuring:

* A title bar matching the app's color theme.
* Question and answer options displayed within cards for a clean, modern look.
* A button to move to the next question, updating the user's score accordingly.

#### Grammar Quiz Screen

The **GrammarScreen** provides multiple-choice questions to help users practice grammar. The design mirrors the vocabulary screens with:

* A title bar consistent with the app's theme.
* Questions and answers displayed in a card format.
* A dynamic scoring system to track user progress.

### **State Management**

The app employs the Provider package for state management, ensuring efficient handling of user progress and scores. The **VocabularyQuizProvider** and **GrammarProvider** classes manage the list of questions, track user answers, and update scores.

#### VocabularyQuizProvider

Manages vocabulary quiz questions, tracks the current question index, and updates scores based on user answers.

#### GrammarProvider

Handles grammar quiz questions, maintains the current question index, and calculates the score.

### **External API Integration**

The Datamuse API is integrated to fetch synonyms, antonyms, related words, and definitions for vocabulary learning. This integration allows the app to provide rich language data without needing an API key, making it accessible and easy to use.

### User Interface Enhancements

The UI is designed to be user-friendly and visually appealing with:

* Consistent color themes and styling.
* Card-like structures for displaying information.
* Interactive elements like buttons and clickable cards for a smooth user experience.

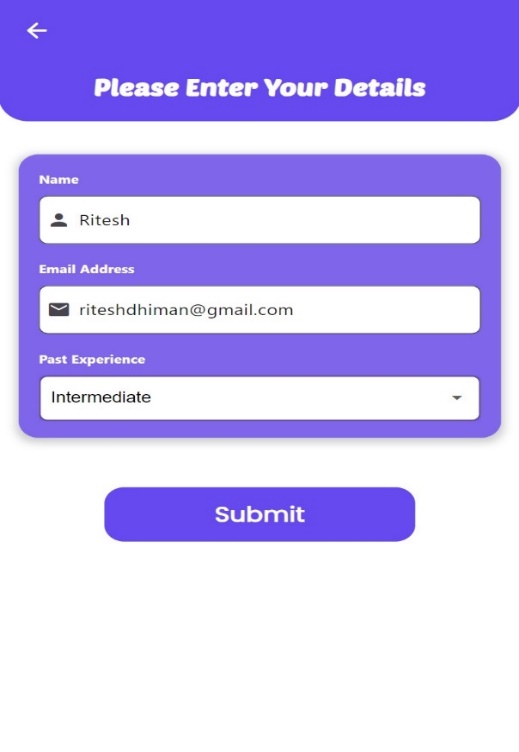
### **Algorithmic Approach**

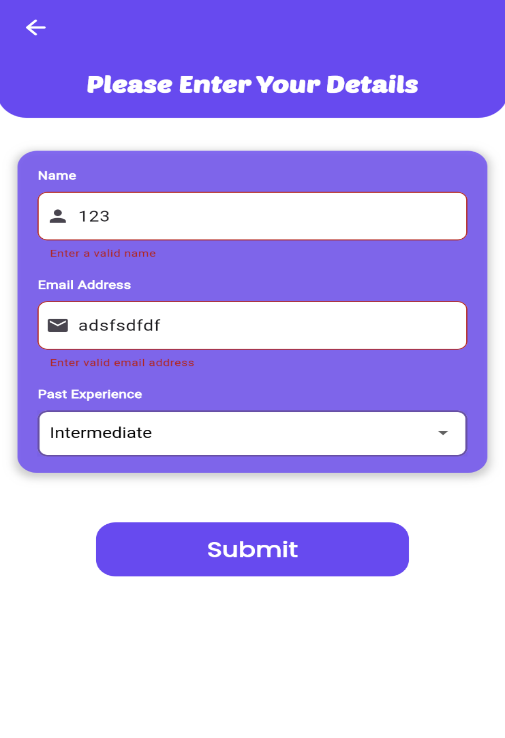
The Language Learning App utilizes several algorithms and systematic approaches to handle different functionalities such as fetching data, managing user progress, and displaying quizzes. Here is an overview of the algorithmic approaches used:

1. **Fetching Vocabulary Data:**
   * Fetch synonyms, antonyms, related words, and definitions from the Datamuse API.
   * Store the fetched data locally for display.
2. **Managing Quiz Questions:**
   * Store quiz questions and options.
   * Track the current question index.
   * Check user answers and update scores accordingly.
3. **User Progress Management:**
   * Use Provider for state management.
   * Update user scores based on their answers.

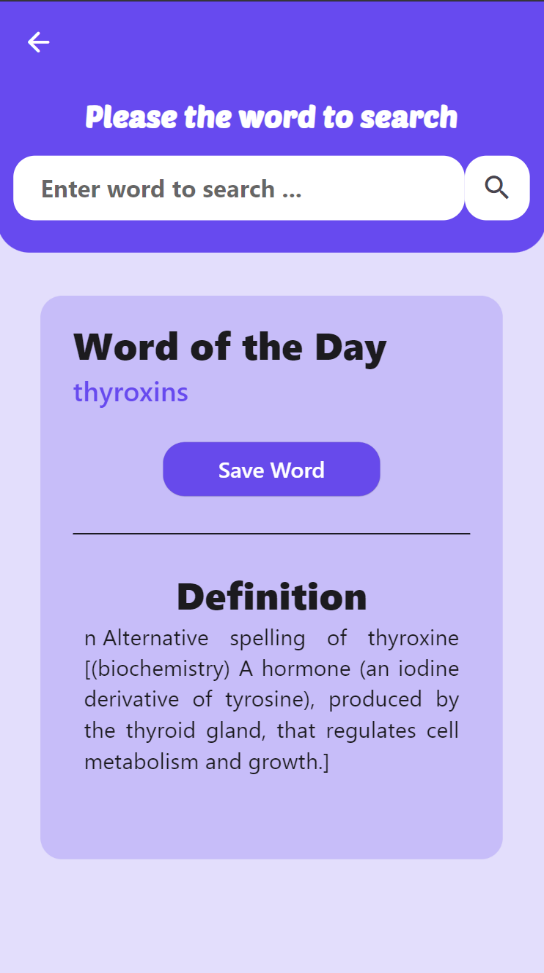
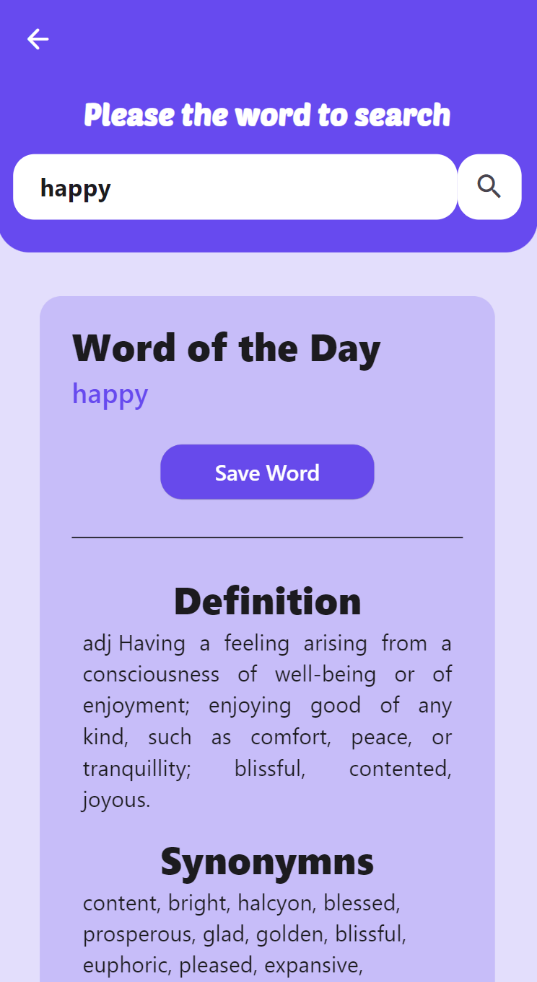
**5. Output (Screenshots)**

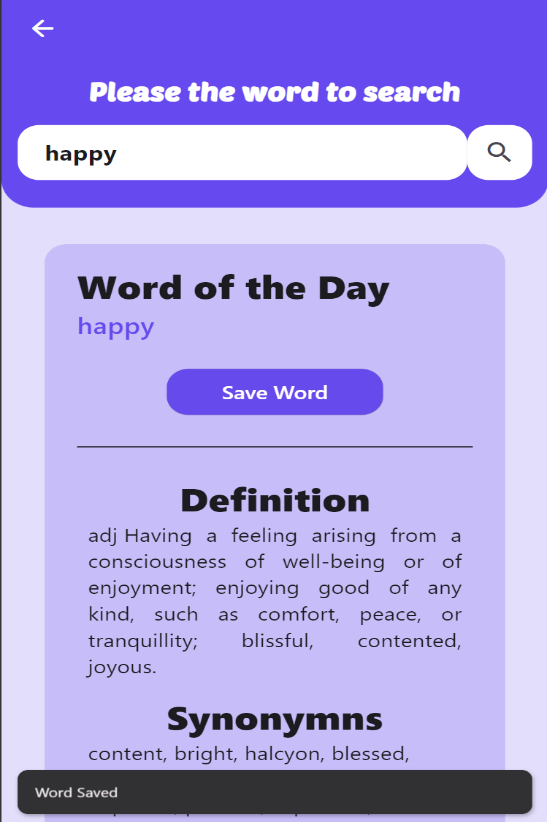
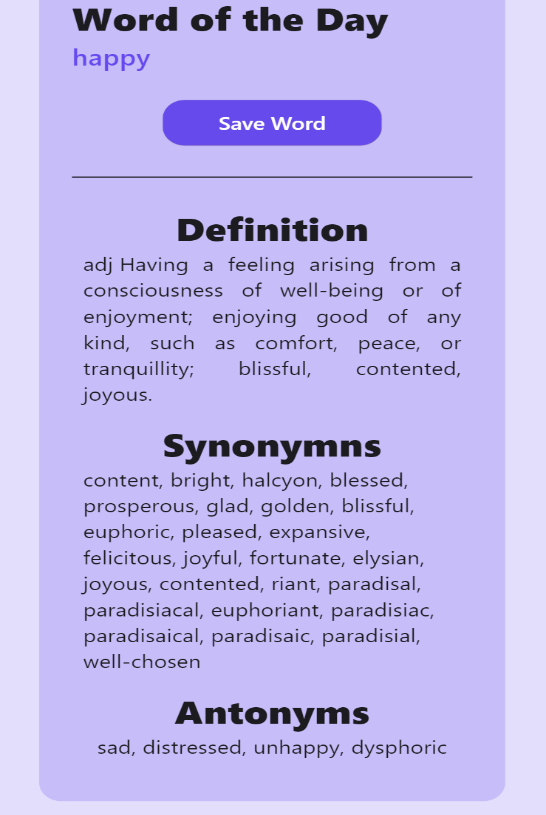
**Getting Started:**



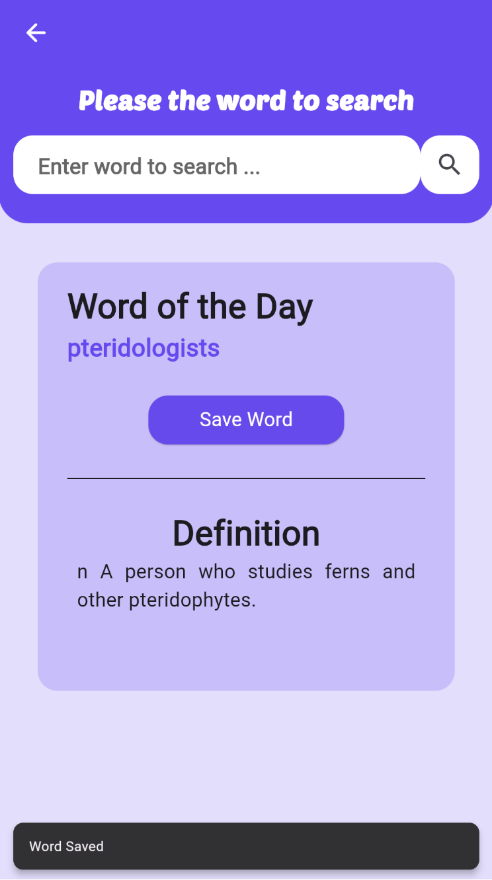
 

**Searching a word:**

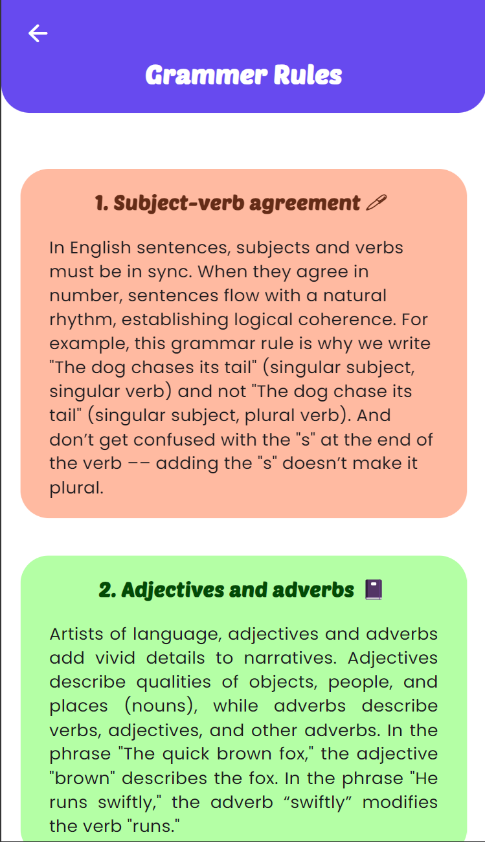
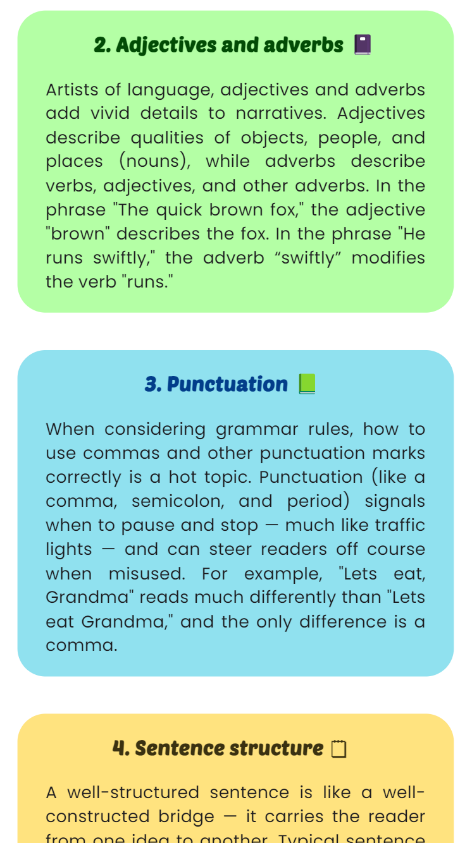
 

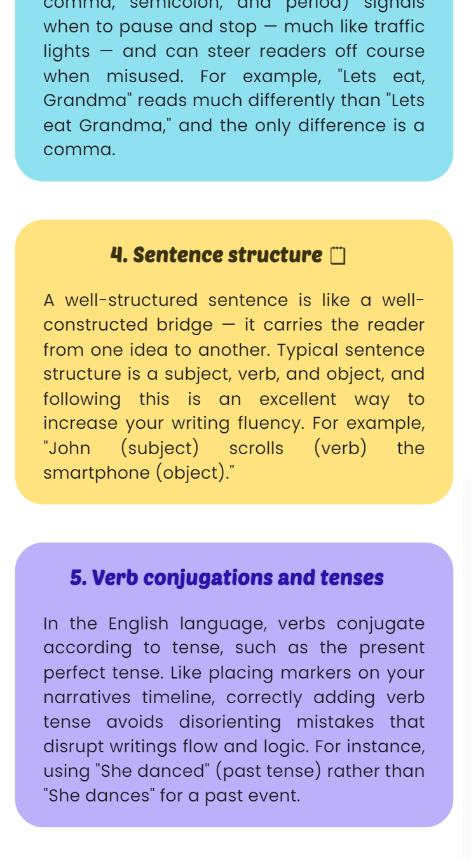
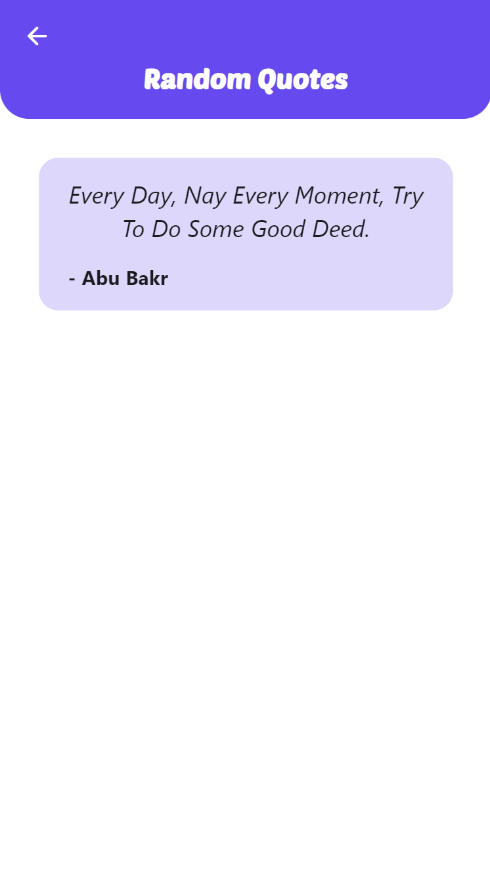
 

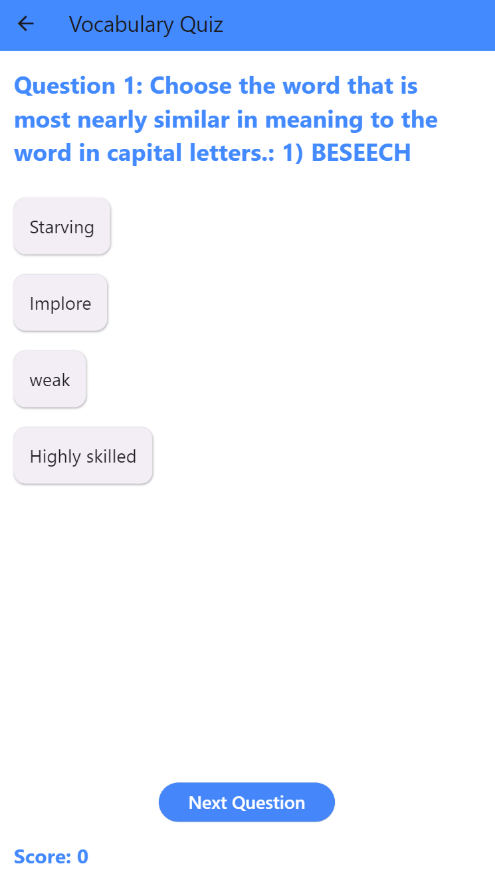
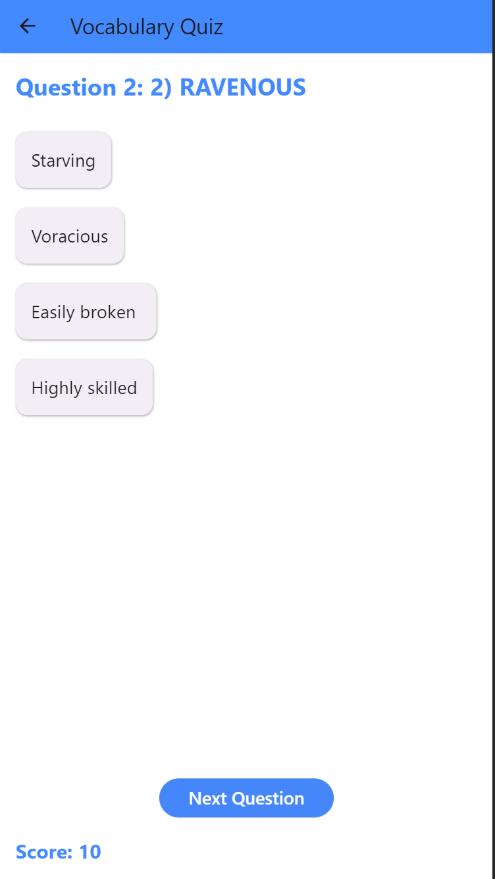
**Saving words:**

**Learning:**

 **Quiz:**

**6. Conclusion and Future Scope**

The Language Learning App is a comprehensive tool designed to enhance vocabulary, grammar, and pronunciation skills. By integrating flashcards, quizzes, and gamification elements, the app provides an engaging and effective learning experience. The use of the Datamuse API enriches the app with extensive language data, while state management with Provider ensures smooth tracking of user progress. Overall, the app combines educational content with interactive features, making language learning both fun and effective.

### **Future Scope**

1. **Enhanced Pronunciation Practice**:
   * Integrate speech recognition for real-time feedback.
   * Add more audio examples and exercises.
2. **Expanded Vocabulary Database:**
   * Include additional language resources and APIs.
   * Provide more contextual examples and usage scenarios.
3. **Adaptive Learning**:
   * Implement machine learning to personalize the learning experience.
   * Dynamically adjust the difficulty level of quizzes.

[**https://github.com/ncu-piha/Flutter-Projects/**](https://github.com/ncu-piha/Flutter-Projects/)

[**https://github.com/RiteshDhiman/LearnIt-App**](https://github.com/RiteshDhiman/LearnIt-App)