Lingo Bingo : A Vocabulary Learning Application

Project Theme

You have to create a fun and interactive language learning application designed to help users expand their vocabulary and improve their communication skills in multiple languages. Learning a new language can be challenging, especially when it comes to retaining vocabulary

Your application will help users to overcome the learning challenge of vocabulary in an easier way. Users can start learning with a simple click. Users need to login to see the vocabularies. It will be handled by Firebase.

You have to create an application for a specific language learning. You can select any language except Bengali and English. Some Language suggestions are given below.

Spanish	Mandarin Chinese	French	German
Japanese	Arabic . In case of arabic word direction will be RTL	Hindi	Portuguese
Russian	Korean	Italian	Turkish

The whole website will be in English language. But json data will be a specific language. For example if you select Japanese as language only your json data will be Japan vocabulary but other sections and pages language will be in English

---

Key Things You must do

GitHub Commits: Include at least 10 meaningful commits with descriptive messages.

Readme.md: Include a README file with the project name, purpose, live URL, key features, and any npm packages you have used.

Responsiveness: Ensure the website is fully responsive on mobile, tablet, and desktop.

Environment Variables: Secure Firebase configuration keys using environment variables.

Unique Design: Create a unique design aligned with your theme. You can use this blog for these kinds of resources

Host your Application: You can choose deployment systems like Netlify, Surge, and Firebase for hosting. As you are developing a single-page application
- ⚠️ensure that the page doesn't throw any error on reloading from any routes.
- ⚠️Add your domain for authorization to Firebase if you use Netlify / surge
- ⚠️ Logged in User must not redirect to Login when reloading any private route

Main Requirements

1. Layout Structure

The whole application will be on a single-page layout. With the following components.

Header: Design a Header for your application.
- Show a welcome message with the user name at the top. (only if user login)
- A header with Logo at the Left/ center, Login Button at the Right
    - If user Logged in (show user image and Log-Out Button)
- A left/center aligned Navbar with following Links
    (Home, start-learning, tutorials, about-us)
- Show 1 more Link if the user logged in (my-profile)

Footer: The design of the Footer is flexible. You have to show Contact information, social media links, and a copyright notice with your own design.

Main: The Main Section will show different pages based on route. We will discuss it below

2. JSON Data Generation

Create a JSON data (array of objects) of a minimum of (50-200) object of vocabularies with the following keys and data types

- **Id:** string   // A unique string
- **word:** string   // original form
- **pronunciation:** string   // pronunciation in English
- **meaning:** string   // meaning in english
- **part_of_speech:** string   // parts of speech
- **difficulty:** string   // easy, medium, difficult
- **Lesson_no:** number   // 1- 10
- **when_to_say:** string   // when to use this word
- **example:** string   //examples with english pronunciation

It is recommended to generate as much as you can.

Don't get confused for website's language. It will be in English . Suppose I have selected Japanese . So my JSON Data will be like this.

```
const japaneseWord = {

 id: "word001",

 word: "ありがとう",

 pronunciation: "arigatou",

 meaning: "thank you",

 part_of_speech: "expression",

 difficulty: "easy",

 lesson_no: 1,

 when_to_say: "Used to express gratitude in informal settings.",

 example: "ありがとう、手伝ってくれて！(arigatou, tetsudatte kurete!) - Thank you for helping me!"

};
```

3. Home-Page

You have to show the following section in the default route.

Banner/Slider Section: Use Language Learning images with a slider.

About Section: Explain the website's mission and how users can learn vocabularies

Success Section: Show User Count, Lesson Count vocabulary count, Tutorial Count. Use Dummy information for Count. Animate this success count using react-count-up

2 extra sections: Add 2 extra sections with your own design.

4. Let's Learn Page

On this Page, users will show the following sections
- Page Title (show the user a Page Title)
- Show users 10 static cards for 10 Lessons. The card will have a simple Text Lesson-1/2/3
    - Clicking the card will redirect you to the lesson/lesson_no page

- Add a Tutorial Section with a YouTube embedded video on **learning the alphabet** for this language
- Add a "View more" button. clicking on It redirects logged in users to the tutorials page.

4. Lessons/:lesson_no `private Route`

This page is only accessible after login; redirect to the login page if the user is not authenticated.

- Show Page Title with Lesson No.
- Show all the vocabularies where the lesson_no is matched with router params.
- Show the vocabularies information in a card format
    - Word, meaning, pronunciation, part_of_speech
    - Color the cards based on the difficulties
    - When to say the button
        - When to say button open a modal with the following data (word, meaning, when to say , example)
- A Button for "Back to Lesson" will navigate users to the let's learn page.

6. Authentication System

On Clicking the Login Button / browsing any private Route user will be redirected to this route.

User Login

User will show a Login page with a form, so that the user can Log in this application.

- Show a Title for Login. & Form with the following fields (Email, Password, Forget Password, Login button)

If the user logs in successfully then navigate him to his desired Route / Home page. If not, show him an error with a toast/error message anywhere in the form.

There will be some other options like

- Show the user a Link for Register so that he can go to the register page.
- Show users a Social Login Button (Google only) . on Clicking it
    - user authenticates with Google
    - Navigate him to his desired Route / Home page.

User Registration

Create a register page with a form, so that the user can register himself in this application.

- Show a Title for registration and a Form with the following fields
  (Name, Email, Photo-URL, Password & Register Button)

If the user registers successfully then navigate him to his Home page. If not, show him an error with a toast/error message anywhere in the form.

Implement password validation For password validation you need to follow the below criteria. Show a password error in the form, and don't Register for an invalid password
- Must have an Uppercase letter in the password
- Must have a Lowercase letter in the password
- Length must be at least 6 character

There will be some other options like
- Show the user a Link for Login so that he can go to the Login page.
- Show users a Social Login Button (Google only) . on Clicking it
  - user authenticates with Google
  - Navigate the user to the Home page.

💡Don't implement email verification method as it will inconvenience the examiner. If you want, you can add these after receiving the assignment result.

⚠️ Logged in User must not redirect to Login on reloading any private route

7. My-Profile(Private Route)   `private Route`

On clicking on my profile at navbar, the user will see this page. The user will see all his Profile Information Here
- Show a welcome Title with the user name.
- Show the user All profile information on this page such as image, email, name, and an update Button which will redirect the user to the update profile route.
- See challenges for updating profile features. will be described at challenge section

8. Tutorial page (Private Route)

- Add Page Title
- Embed 6-8 videos on this language from youtube. Make sure your video is playing.
- A Button "Learn Vocabularies" will navigate users to the "start-lesson" route.

9. About Us (optional)

- Write about yourself , your other project info , skills

Error Page

Add a 404 Page: If the user wants to visit invalid routes Show the user a Not Found Page with a button so that the user can go back to the home route.

Challenges

1. Implement any One Animation package on the Home Page

- [AOS Package](#),
- [Animate.css](#),

2. Forget Password

Make your forgot password feature functional. Redirect the user to the forget password route by clicking forget password, which will contain a form with an email field and a reset password button.

- If a user fills in the email input on the login page. Show the email also in the forget password form.
- On clicking the reset button. Redirect the user to Gmail.

3. Update Information Feature

In my-profile route, there will be an update button on Clicking it. Take the user to update profile route

- Show the user a form with 2 input fields (photo-URL and Name)
- An Update Information button.

On Clicking update, update the user profile from Firebase and navigate the user to the my profile Route

4.   Speak up your Vocabularies

On clicking a particular vocabulary card, pronounce the word so that the user can hear the vocabularies. Use this Link https://codepen.io/Ferdous-Zihad/pen/QWeoRMo to get ideas. Your task is to implement it on a React project.

NOTE :

If you use Spanish , make sure the Spanish language with voice install on your PC.

Go to PC Settings > Language > install this language with voice pack

⚠️ Before you submit, check carefully-> you did the things mentioned in the '[Key Things You Must Do](#)"

What to Submit

- **Assignment Category:**
- **GitHub Repo Link:**
- **Live Link:**