Translate App

Task:

Build a single-page Vue.js application that translates English text to Hindi using the Google Translate API. The application should also display a history of the last 10 translated words/phrases.

Requirements:

1. **User Interface:**
   - A text input field for the user to enter English text.
   - A button to trigger the translation.
   - A display area to show the translated Hindi text.
   - A list to display the last 10 translated words/phrases (English and Hindi).
2. **Functionality:**
   - When the user clicks the translate button, the app should send the English text to the Google Translate API.
   - The API response (Hindi translation) should be displayed in the designated area.
   - The translated word/phrase should be added to the history list.
   - The history list should maintain a maximum of 10 entries, removing the oldest entry when a new one is added.
3. **Technical Requirements:**
   - Use Vue.js for building the application.
   - Use the Google Translate API(https://cloud.google.com/translate) or any other public api (like https://mymemory.translated.net/doc/spec.php)   for translation.
   - Implement proper error handling for API requests (e.g., display an error message if the API is unavailable).
   - Ensure the code is well-structured, readable, and maintainable.
   - Consider using appropriate Vue.js features like components, props, and data binding.

Bonus Points:

- Implement additional features like:
  - Allowing the user to clear the history.
  - Providing options for selecting different languages.
  - Implementing a "speak" functionality to pronounce the translated text.
- Write unit tests for the application logic.

Evaluation Criteria:

- Functionality: Does the application meet all the requirements?
- Code quality: Is the code well-structured, readable, and efficient?
- Error handling: Are potential errors handled gracefully?
- User experience: Is the application intuitive and easy to use?
- Bonus features: Are any additional features implemented?

Submission Guidelines:

1. **Code Repository:**
    - The candidate should create a public Git repository (e.g., on GitHub, GitLab, Bitbucket) to host their code.
    - The repository should include a README file with clear instructions on how to run the application.
    - Commit code regularly with clear commit messages that describe the changes made.
2. **Deployment (Optional):**
    - Deploy the application to a publicly accessible platform (e.g., Netlify, Vercel, Heroku).
    - Include the deployment URL in the README file.
3. **Submission:**
    - Submit the link to the Git repository via the designated platform (e.g., email, coding platform submission form).
    - Ensure the repository is accessible to the reviewers.

Tips for the Candidate:

- Demonstrate your understanding of Vue.js concepts and best practices.
- Show your ability to interact with external APIs.
- Focus on writing clean and maintainable code.
- Consider the user experience when designing the interface.
- Don’t commit the API key in the repository.