IDEA DE PROGRAMA

Elección de una aplicación que no sabemos hacer, al menos por el momento.

Entre corchetes [ ] y en otro tono de color están los elementos que por el momento son más opcionales

Programa

Programa en el que *[mediante la api de ChatGPT]* se genera un texto de un número determinado de palabras que el usuario deberá escribir lo más rápido posible y de manera correcta, al terminar se guardará en una base de datos la velocidad con la que se realizó, cada letra errónea será 1s más al tiempo y se mostraría de la siguiente manera: Tiempo= 00:10//Errores: 5//Final: 00:15. Esto se mostrará un ranking con los usuarios y sus mejores tiempos.

Permitirá la personalización visual del programa, por ejemplo, un modo oscuro y un modo claro, color del texto generado, tamaño, fuente… *[Los temas del texto podrán ser enviados por el usuario o utilizar uno aleatorio elegido de entre varios temas predeterminados, si no se puede realizar la conexión a la API de ChatGPT,*] también se podrán usar textos predeterminados para permitir el uso offline, estos textos predeterminados podrán ser enviados por los usuarios y después permitidos por un administrador.

El uso de una cuenta solo será necesario si se quiere guardar los tiempos realizados.

Ámbito de la aplicación

Uso personal, entretenimiento, competitividad…

Objetivos

Usuario estándar

Personalizar su perfil, hacer uso del programa de manera online/offline…

Usuario empresa

Administrar los usuarios registrados desde el cliente y permitir las peticiones de texto

Hardware

Windows *[y Linux]*

Usabilidad

- Preguntar al usuario sobre (nombre, apellidos, email, usuario, fecha nacimiento…)
- Competencias y habilidades.

Interacción

Al abrir el programa ver los rankings y entrar a jugar o administrar su cuenta, ver información sobre el programa y contacto...

Requisitos

1. Requisitos funcionales (el usuario se ha de registrar, control de usuarios, creación de una cuenta, comprobaciones de cuentas, contraseñas deban cumplir un patrón…)
2. Requisitos sobre interfaces de usuario. Por ejemplo, la interfaz tendrá una interfaz sencilla, agradable y fácil de entender y utilizar. Tendrá un diseño minimalista y un menú de navegación con el que poder desplazarse entre las diferentes funciones.
3. Requisitos de rendimiento (tiempo de respuesta de cada acción).
4. Requisitos de seguridad. Por ejemplo, las contraseñas ingresadas por los usuarios deben de cumplir una serie de requisitos:

- Debe contener al menos un carácter especial (! @ # $ % & / () = + ? [ ] ~ – ^).
- Debe contener al menos un carácter mayúscula y otro minúscula.

Debe de tener una longitud mínima de 8 caracteres.

Para ser almacenadas las contraseñas en la base de datos, debe aplicarse un cifrado hash sobre ellas antes de guardarlas.

Implementación de una capa más de seguridad para la contraseña del usuario mediante la autenticación de doble factor.
Esta autenticación puede ser de dos tipos: Código de verificación mediante email o aplicación externa (Google Authenticator o similares).

5. Requisitos de disponibilidad o fiabilidad (BBDD):
    a. La base de datos debería estar siempre disponible, en caso de no estarlo o no tener conexión podrá utilizar el programa de manera offline, pero sin guardar puntuaciones. Si la base de datos no está disponible mostrar un aviso en el programa.
6. Requisitos de recuperación. ¿Qué ocurre cuando el sistema falle? ¿Se pierden todos los datos?
    a. Habrá una copia de seguridad periódica de la base de datos.

Bocetos de diseño

En la página puedes hacer click en los siguientes botones para ver el resto de las partes del programa: Jugar, Inicio, Información, Intro para terminar y en la pantalla de inicio en el Usuario

Página con el diseño hecha en FluidUI:

https://www.fluidui.com/editor/live/preview/cF9pYnRBem54azZzek9DRjJ3ZHBtY0ZPYk5Cb0w3enh0Nw==