2º Trimestre – EXAMEN de Control

Temas tratados: Vue (slots, composable, Pinia, fetch), I18n, Router, Typescript y JQuery
Adjunto proyecto: BBDD Mongodb en Docker para el login

Blog con publicación dinámica

Índice

Descripción

La tarea consiste en ampliar el portfolio para incluir una sección de blog dinámico.

Deberás implementar un sistema de publicaciones que permita listar, ver, crear y editar entradas de blog utilizando las tecnologías indicadas.

Instrucciones detalladas

1. Configurar el Router

- Crea una nueva ruta `/blog` que liste todas las publicaciones.
- Configura rutas dinámicas para cada entrada del blog, por ejemplo: `/blog/:id`. Esta ruta debe mostrar el contenido completo de una publicación.

2. Crear el componente BlogList.vue

- Este componente debe mostrar una lista de publicaciones.
- Utiliza **fetch (como alternativa pudes usar Axios)** para obtener las publicaciones iniciales desde una API o un archivo JSON proporcionado (simula los datos).
- Integra un **slot** para personalizar el encabezado de la lista.
- Añade un botón que permita navegar a un formulario para crear nuevas publicaciones.

3. Crear el componente BlogPost.vue

- Este componente debe mostrar el contenido completo de una publicación seleccionada (usando la ruta dinámica `/blog/:id`).
- Asegúrate de que los datos de la publicación se obtengan dinámicamente utilizando **fetch**.

4. Crear el componente BlogForm.vue

- Este componente debe permitir crear y editar publicaciones.
- Implementa un formulario con validaciones usando **TypeScript** (campos obligatorios: título, contenido y autor).
- Al enviar el formulario, los datos deben guardarse en Firebase (Firestore).
- Al editar, los datos existentes deben cargarse automáticamente en el formulario.

5. Usar Pinia para la gestión de estado

- Implementa un store en Pinia para gestionar las publicaciones de manera centralizada.
- El store debe incluir acciones para cargar, añadir y editar publicaciones.

6. Soporte multilingüe con I18n

- Integra traducciones dinámicas en todos los textos visibles de los componentes (por ejemplo: "Crear publicación", "Editar", "Guardar", "Título", "Contenido").

7. Añadir animaciones con jQuery

- Usa jQuery para animar la aparición de nuevas publicaciones en la lista.
- Añade un efecto de transición al mostrar el contenido de una publicación en BlogPost.vue.

8. Autenticación con MongoDB en Docker

- Restringe la creación y edición de publicaciones para que solo los usuarios autenticados puedan acceder al formulario.
- Asegúrate de manejar el inicio de sesión y cierre de sesión en MongoDB

9. Documentación y limpieza del código

- Mantén una estructura limpia y organizada del proyecto.
- Comenta tu código.
- Incluir un archivo README.md que explique cómo probar tu solución y describa las decisiones técnicas tomadas durante el desarrollo

Rúbrica

Criterio	Puntos
Router: Implementación de la ruta `/blog` correctamente.	0.5
Router: Implementación de la ruta `/blog/:id` correctamente.	0.5
BlogList.vue: Lista de publicaciones funcional con fetch o Axios.	0.75
BlogList.vue: Uso de slot.	0.25
BlogPost.vue: Visualización del contenido dinámico con fetch.	1.0
BlogForm.vue: Creación y edición de publicaciones con validaciones.	1.5
Pinia: Gestión centralizada de publicaciones en el store.	1.0
I18n: Soporte multilingüe completo en los textos visibles.	1.0
jQuery: Animaciones al añadir publicaciones y mostrar contenido.	1.0
Autenticación: Restricción de acceso al formulario mediante autenticación.	1.5
Código limpio y comentado: Código organizado, sin redundancias y con comentarios explicativos claros.	1.0