# A picture containing drawing Description automatically generated

ARQUITECTURA DE APLICACIONES WEB (SI705)

# SIMULACRO Examen PARCIAL

# 2022-2

# Sección: Todos

# Profesores: Todos

**Duración:** 110 minutos

**Indicaciones:**

Sobre entrega del proyecto de software y backend fake (json server).

1. El examen consta de 1 pregunta, y tendrá 110 minutos para resolverlas.
2. La pregunta es de tipo Proyecto de Software y la entrega de su respuesta es a través del envío de los siguientes archivos :

* Debe crear una carpeta con la siguiente nomenclatura upc-pre-codigoalumno-proyecto
* Luego colocar los siguientes archivos en la carpeta
  + upc-pre-codigoalumno-proyecto.zip: proyecto solución
  + upc-pre-codigoalumno-imágenes-solucion.pptx: imágenes de evidencia de los resultados obtenidos, en cada imagen debe aparecer con camara prendida.
* Finalmente, debe subir el archivo la carpeta en formato .zip en la actividad de entrega del examen parcial.

Donde codigoalumno debe reemplazarlo por su respectivo código alumno.

Recuerde eliminar la carpeta node\_modules antes de subir el proyecto. Caso contario se penalizara con -4 ptos a la nota obtenida.

1. Para el desarrollo del examen se requiere archivos de configuración que se encuentran como archivo adjunto (**upc-pre-202202-si705-simualcro-examen-parcial\_files.zip**). El cual debe ser utilizado con json-server (<https://www.npmjs.com/package/json-server>) para simular un backend con las características indicadas.

La información de los Knowledge Items se encuentra en el endpoint: http://localhost:3000/knowledges

Sobre consultas al equipo académico y reporte de problemas técnicos

4.      Cada examen cuenta con un equipo académico, el cual estará conectado durante los primeros **15 minutos del examen**.

5.      El alumno debe dedicar **los primeros 15 minutos** a revisar las preguntas del examen y de presentarse alguna duda enviar un correo al(los) profesor(es)

6.      De no recibir respuesta del equipo académico, o tener algún inconveniente adicional **pasado los primeros 15 minutos**, puede comunicarse con la profesora

7.      Los profesores en mención, solo recibirán correos provenientes de las cuentas UPC, de ninguna manera se recibirán correos de cuentas públicas.

8.      Ante problemas técnicos, debe de forma obligatoria adjuntar evidencias del mismo, como capturas de pantalla, videos, fotos, etc. Siendo requisito fundamental que, en cada evidencia se pueda apreciar claramente la fecha y hora del sistema operativo del computador donde el alumno está rindiendo el examen.

9.       Los problemas técnicos se recibirán como **máximo 15 minutos culminado el examen**.

**Enunciado:**

**Pregunta 1 (20 p.).**

**Wasteless Food, inc.**

Su cliente, Wasteless Food, inc., está embarcado en la misión de inspirar y empoderar a las personas a tomar acciones en contra del desperdicio de comida. Para ello, su equipo se encuentra desarrollando para wastelssfood.org, una plataforma para brindar soporte a sus operaciones, sin embargo, desea ir iniciando con el desarrollo de software para la experiencia web.

Para el desarrollo web de lado web frontend, se ha seleccionado TypeScript como lenguaje de programación y Angular como Frontend Framework.

Se le encarga el desarrollo de una aplicación web que implemente las siguientes características:

* El nombre del proyecto debe ser **knowledges-app-codigoalumno.** Donde código alumno debe ser reemplazado por tu código.
* Un toolbar, donde a la izquierda se muestra el branding **Knowledges App** y a la derecha se muestra las opciones “Home” y “Knowledges”. (ver imagen de anexo 1)
* La vista Home muestra un card indicando “Current Knowledges Count” junto con un valor numérico indicando la cantidad actual de Knowledges actualmente registradas. (ver imagen de anexo 2)
* La vista List Knowledges presenta una vista tabular con soporte de operaciones CRUD, para mostrar todos los knowledge disponibles. (ver imagen de anexo 3)

La vista **Home** es accesible desde la ruta /home.

La vista **raíz** (accesible desde la ruta de navegación tanto /) debe redirigir al usuario a la vista /home.

En la vista **List** **Knowledge**s (accesible en la ruta business/**knowledge**s), cada elemento de la tabla muestra la información de un **knowledge**, incluyendo el title, urlToImage y summary, contente según las especificaciones de material design. Debe permitir, además Agregar, Actualizar o Eliminar **Knowledge**s. Para cada fila se muestran controles con íconos para editar y eliminar.

La vista **New Knowledge** (accesible en la ruta de navegación admin/knowledge/new)

Debe presentar un formulario de material design que permita registrar un Kownledge Item.

Para el ingreso se usa input fields para las características del ítem, incluyendo: title, urlToImage, summary, content. Considerar como obligatorios los campo title, urlToImage y summar. El title debe tener máximo 60 caracteres. Considere mensajes de error ante el incumplimiento de reglas en cada input que corresponda y placeholders para brindar mayor ayuda al usuario. Debe incluir dos botones: Cancel y Save. Cancel redirige a la vista List Knowledge sin registrar información, mientras que Save registra el Knowledge Item, para luego redirigir al usuario a la vista List Knowledge, donde debe apreciarse el nuevo knowledge item.

La vista **Edit Knowledge** (accesible en la ruta de navegación admin/knowledge/edit/:id)

Debe presentar un formulario de material design que permita actualizar un Kownledge Item. Debe incluir dos botones: Cancel y Update. Cancel redirige a la vista List Knowledge sin registrar información, mientras que Update actualiza el Knowledge Item seleccionado, para luego redirigir al usuario a la vista List Knowledge, donde debe apreciarse el knowledge item editado.

**Restricciones técnicas:**

* El equipo requiere que la interfaz de usuario esté basada en **Angular Material**, mientras que para la comunicación con el backend debe apoyarse en **HttpClient** (incluido en @angular/common/http).
* La aplicación debe utilizar **@angular/router** para el manejo de routing en la aplicación. Debe utilizar componentes de angular material MatSliderModule, MatToolbarModule, MatTableModule, MatFormFieldModule, MatInputModule, MatPaginatorModule, MatIconModule, MatButtonModule, MatSnackBarModule, MatCardModule. Los componentes no deben estar importados en app.module.ts sino que debe crear un module exclusivo para componentes angular material.
* El nombre del proyecto deber ser **knowledges-app-codigoalumno. Debe reemplazar codigoalumno por su respectivo código de alumno.**
* La estructura de carpetas debe ser de acuerdo con lo explicado en clase.
* Las rutas (path) deben tener el nombre que se indica en el enunciado.

Rúbrica de calificación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criterio de Calificación** | **Excelente** | **Promedio** | **Deficiente** | **Calificación** |
| **C01. Building y ejecución** | Al abrir el proyecto y ordenar la ejecución, ésta se inicia sin problemas. La aplicación es accesible en la ruta indicada.  Cumple con todas las restricciones técnicas | La aplicación no llega a iniciar y ejecutarse, sin embargo el proceso de building llega a concluir. Cumple parcialmente con las restricciones técnicas. | Al cargar el proyecto el proceso de building presenta errores y no llega a concluir o no cumple con las restricciones tecnicas. |  |
|  | **3.0 puntos** | **1.5 punto** | **0 puntos** |  |
| **C02. User Interface & Home View** | La interfaz de usuario cumple con las características solicitadas para la estructura, elementos de la interfaz de usuario y rutas de navegación. Se evidencia que la aplicación recupera y presenta en Home el card con lo solicitado en base a la información del backend. | La interfaz de usuario no cumple con todas las características solicitadas, sin embargo, recupera y presenta en Home la información solicitada del backend. | La aplicación no implementa una interfaz de usuario o no presenta la información indicada. |  |
|  | **6.0 puntos** | **3.0 puntos** | **0 puntos** |  |
| **C03. Knowledge View and Operation** CRUD | Se evidencia que la aplicación proporciona la opción Knowledge, que lleva a la vista que incluye un Data Table y elementos de UI para mantener la información de Knowledges, soportando consulta, adición, actualización y eliminación, aplicando validaciones, control de errores y comunicación de mensajes para el usuario. La vista cumple con los requisitos de interfaz de usuario y ruta de navegación. | La aplicación permite mantener de forma parcial Knowledges, soportando solo algunas de las operaciones CRUD, o aplica de forma parcial validaciones, o controla de forma parcial los errores o no comunica de forma adecuada al usuario. | La aplicación presenta la vista Knowledges, pero no se puede realizar operaciones CRUD sobre Knowledges, o no se implementa la opción. |  |
|  | **7.0 puntos** | **3.5 puntos** | **0 puntos** |  |
| **C04. Code Organization** | El desarrollador organiza el código y los elementos de frontend de la solución, aplicando buenas prácticas de TypeScript y Angular, agrupando los elementos de la solución según convenciones, manteniendo organización de paquetes y carpetas recomendadas de acuerdo con el demo desarrollado en clase. | El desarrollador aplica parcialmente en frontend convenciones, recomendaciones y buenas prácticas de TypeScript o Angular. | No se evidencia un criterio de organización  para los elementos de la solución, limitándose a la estructura por defecto. |  |
|  | **3.0 punto** | **1.5 puntos** | **0 puntos** |  |
| **Total** | **20 puntos** | **10 puntos** | **0 puntos** |  |

Anexo

Logo

Description automatically generatedImagen 1

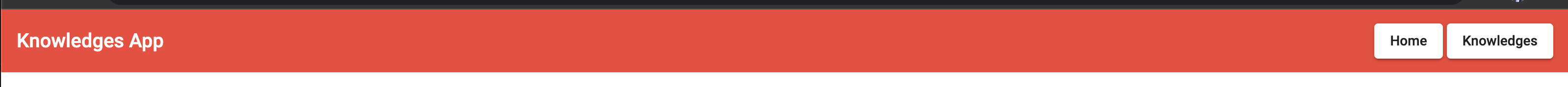


Imagen 2

Logo

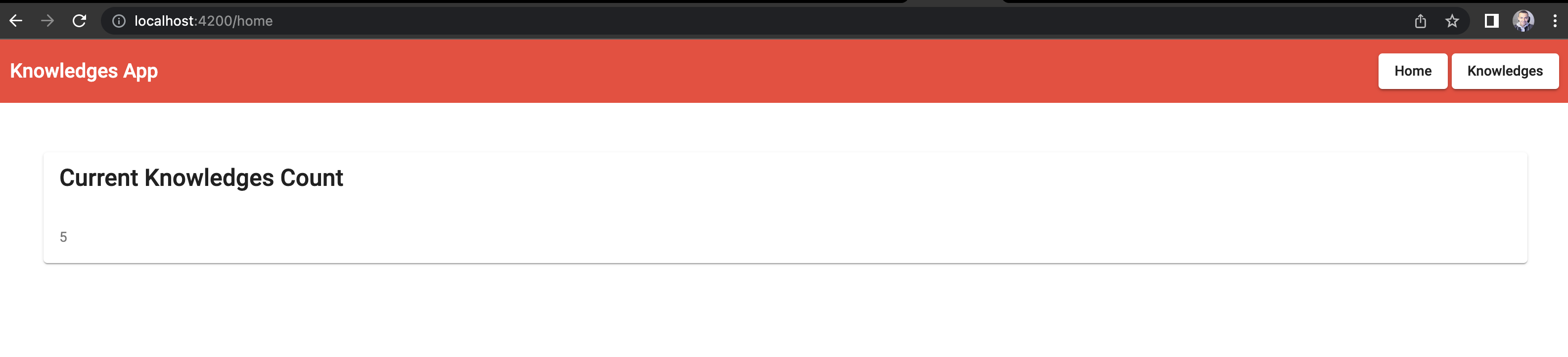
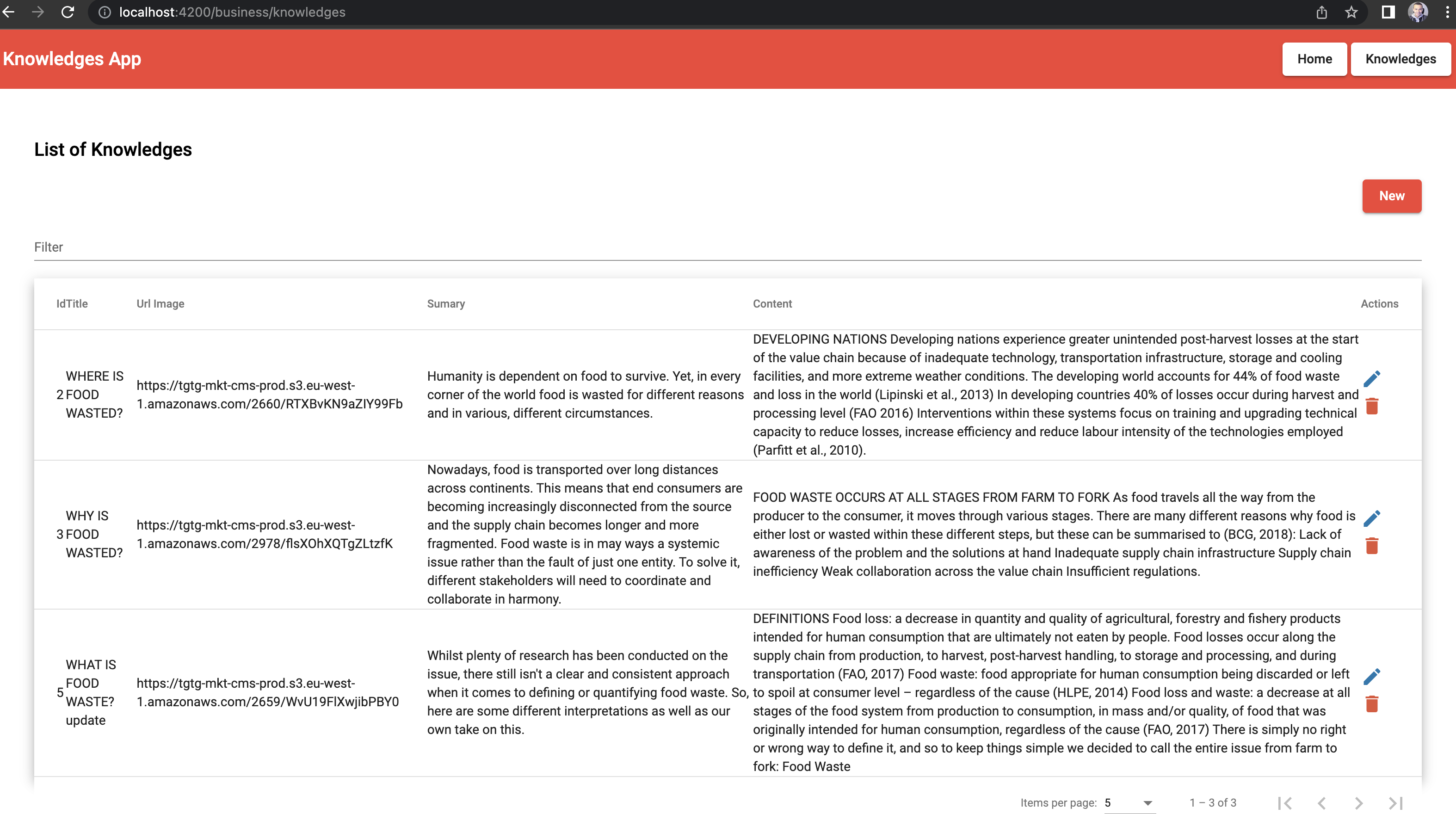
Description automatically generated

Imagen 3

Logo

Description automatically generated

Logo

Description automatically generatedGraphical user interface, application, shape, Word, arrow

Description automatically generated

Logo

Description automatically generatedShape, arrow

Description automatically generated