



Diseño web dinámico Oto Zra

Fecha: 7 de agosto de 2024

Integrantes:

Howlin Ignacio

Gustavo Benitez

Gabriel Solotorevsky

Síntesis

Una página web que obtiene, maneja y permite el acceso a libros, en forma de secuencias de imágenes, que envían los docentes registrados a través de una página web.

Objetivo

Se busca resolver el problema de el acceso a la información educativa en las escuelas, proponiendo un lugar centralizado y privado donde los docentes podrán publicar sus materiales para que lo puedan acceder sus alumnos sin tener que pasar por una tercera aplicación o comprar los materiales físicos.

Desarrollo

El servidor principalmente obtendría archivos del profesor que serán guardados en una carpeta llamada "Media" para que luego puedan ser referenciados por la publicación que hace uso de estos archivos por su URL, los cuales luego serán embebidos en la página, esto hace que el archivo se muestre como el navegador decida.

En referente al inicio de sesion y registro, perteneceria a una url separada a las demas y una vez realizado con exito se guarda la sesion en una cookie que se mantiene en el navegador hasta cerrar sesion. Todos los que entren en la pagina seran estudiantes, pero solamente los que tienen permisos del administrador pueden iniciar como profesor (Con usuario y contrasena) y estos usuarios son los unicos capaces de publicar libros. El unico capaz de crear cuentas de profesores es el administrador.

Funciones

Login especifico y determinado por el administrador para los profesores y se almacena la sesion a travez de cookies

Publicacion y obtencion de publicaciones y archivos

Interfaz web

Urls

E = Entrada, lo que se espera del usuario

S = Salida, lo que devuelve el servidor

<x> = Una variable en la URL

/x = Una url que utilizara la entrada y salida

(GET)/(POST)/(DELETE) = Metodo que usa la URL, servirian como filtros para ciertas operaciones

/(GET)

E: -

S: HTTP_PAGE('Main')

/publicacion_page<id> (GET)

E: id: int

S: HTTP_PAGE('Publicacion')

/publicacion_page (POST)

E: Cookie: ID

S: HTTP_PAGE('Publicar')

/login_page (GET)

E: -

S: HTTP_PAGE('Login')

/publicacion<id> (GET)

E: id: int

S: {Titulo
Descripcion
Categoria
Fecha
Archivo
Portada}

/publicacion<cant> (GET)

E: cant: int

S: Array[Publication_id]

/publicacion<cant><order> (GET)

E: {cant: int
order: String}

S: Array[Publication_id]

/publicacion<cant><type><filtro> (GET)

E: {cant: int
type: String
filtro: String}

S: Array[Publication_id]

/publicacion&view=true<id> (GET)

E: {view: bool
id: int}

S: {Titulo
Descripcion
Portada}

/publicacion (POST)

E: {Titulo: String
Cookie: ID
Descripcion: String
Archivos: Files
Portadas: File}

S: HTTP_REDIRECT('/publicacion&id=\$id')

/buscador (GET)

E: -

S: HTTP_PAGE('Buscador')

/buscador (POST)

E: Entrada: String

S: Array[Publicacion_id]

/login (GET)

E: Cookie: ID

S: Nombre: String | Err

/login (POST)

E: {Email: String
Contraseña: String}

S: Cookie: ID | Err

/categoria (GET)

E: -

S: Array[Categoria]

/login (DELETE)

E: Cookie: ID

S: Result: Ok | Err

/autores (GET)

E: -

S: Array[Autores]

Base de datos / Estructura:

Usuario:

nombre: string
gmail: string
contrasena: string
publicaciones: many(key(publicacion))

Publicacion:

titulo: string
descripcion: string
categoria: many(key(categoria))
fecha: time(now)
archivos: many(key(media))
portada: key(media)

Categoria:

nombre: string

Media:

archivo: filefield(upload_to='/media')

Diagrama de paginas / subpaginas:

Cuadrados = URLS

Capsulas = Botones / Elementos de la pagina

