Challenge

El componente de Booking, debe contener un formulario de busqueda con destino, origen, fecha inicio, fecha de fin, clase de la cabina, etc. Esta informacion del formulario de busqueda se debe de cargar de un listados definidos que pueden esta guardado en BD o en un archivo ya que esa informacion no varia frecuentemente, yo recomiendo que todo lo estatico o casi está tico se guarde en ficheros.

Para tener informacion de los vuelos programados las compañias deben entrar al sistema y hacer la programacion de dichos vuelo con la informacion correspondiente, mediante otra funcionalidad.

Al buscar se debe enviar todos los parametros necesarios al servidor y realizar la busqueda de los vuelos programados, esa informacion debe encontrarse en la BD o se puede consumir una API de un tercero que brinde dicha informacion. El resultado de la busqueda se muestra en el listado de vuelos organizada y con toda la informacion posible: hora inicio, hora de fin, clase, precio, compañía, etc. Se brinda la opcion de que se pueda reservar(Book), se mostraria un modar para confirmar la informacion del vuelo, los datos del pasajero y los datos de pagos donde se realizaria en una seria de pasos las acciones correspondientes. Una ves reservado, se le brinda la necesaria al usuario de la reserva del vuelo y se guarda en el historico para darle seguimiento o permitir cancelar la reserva.

Se codificó la carga de datos del formulario de busqueda y la busqueda de los vuelos

**Tecnologías empleadas**:

Boostrap 5.3, Vuejs 2.5, CodeIgniter 4.x, Postgres, Arquitectura MVC.

**Estilo de código**:

* 80 caracteres por lineas.
* 4 espacios para indentado.
* Las clases se declaran en StudlyCaps.
* Los metodos y propiedades se declaran en camelCase y tienen que llevar la visibilidad.
* Las constantes se declaran en mayuscula y separadas por underscope.
* Los metodos y clases deben estar documentados y los parametros de los metodos debe tener declarado el tipo, así como el retorno de cada metodo.
* Para los strings que no contengan variables se debe usar ‘’.
* La declaracion de array es con [].
* Utilizar elseif en ves de else if.
* Las comparaciones siempre deben ser estrictas y no se debe asignar variables dentro de una condicion.
* Etc.

**Seguridad y validaciones**:

* El sitio debe estar en https.
* Se debe crear un virtual host y configurar el framework que permita el hostname.
* Configurar el framework para las politicas de seguridad de contenido.
* Las peticiones para cargar informacion de los filtro se deben hacer por GET.
* Las peticiones para guardar hacer la reserva y buscar los vuelos se hace por POST.
* Comprobar que las peticiones sean de tipo AJAX para las acciones que las llevan.
* La informacion que se consulta o guarda en BD se debe filtrar y escapar antes de cualquier consulta SQL.
* Se debe validar en la vista y en el servidor.
* La informacion que se guarda debe pasar por las reglas de validacion definidas en el framework para cada campo.
* Enmascarar las peticiones a las acciones de cada controlador definiendo nuevas rutas.
* Etc.

**Base de datos**:

* El diseño de la bd debe tener un equilibrio entra rendimiento, volumen de informacion y la 3era forma normal.
* Cada tabla se debe analizar por separado y revisar nivel de operaciones de lectura y escritura, tambien revisar el crecimiento en el tiempo.
* Las peticiones siempre deben tener filtros y limites, para no estresar el servidor.