Challenge

The Booking component must contain a search form with destination, origin, start date, end date, cabin class, among others. This information from the search form must be loaded from a defined list that can be saved in a DB or in a file, since this information does not change frequently, I recommend that everything that is static or almost static be saved in files.

In order to have information on scheduled flights, companies must enter the system and schedule said flights with the corresponding information, through another functionality.

When searching, all the necessary parameters must be sent to the server and the search for scheduled flights must be carried out. This information must be found in the DB or an API from a third party that provides said information can be consumed. The search result is shown in the list of flights organized and with all possible information: start time, end time, class, price, company, etc. The option that can be reserved (Book) is provided, a modal type form would be displayed to confirm the flight information, passenger data and payment data where the corresponding actions would be carried out in a series of steps. Once reserved, the necessary information is provided to the user of the flight reservation and it is saved in the history to follow up or allow the reservation to be cancelled.

The data load of the search form and the search for flights were coded, missing the booking of flights and paying for the reservation. For that, the model classes previously mapped with the tables in the database would be used, where the information would be recorded, previously the defined rules and validations would be applied.

**Technologies used**:

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

**Code Style**:

* 80 characters per line. - 4 spaces for indentation. - Classes are declared in StudlyCaps.
* The methods and properties are declared in camelCase and have to carry visibility.
* Constants are declared in uppercase and separated by underscope.
* The methods and classes must be documented and the parameters of the methods must have the type declared, as well as the return of each method.
* For strings that do not contain variables, use ''.
* The declaration of array is with [].
* Use elseif instead of else if.
* Comparisons must always be strict and variables must not be assigned within a condition.
* Etc.

**Segurity y validations**:

* The site must be in https.
* You must create a virtual host and configure the framework that allows the hostname.
* Configure the framework for content security policies.
* Requests to load filter information must be done by GET.
* Requests to save the reservation and search for flights are made by POST.
* Check that the requests are of the AJAX type for the actions that carry them.
* The information that is consulted or saved in the DB must be filtered and escaped before any SQL query.
* It must be validated in the view and in the server.
* The information that is saved must go through the validation rules defined in the framework for each field.
* Mask the requests to the actions of each controller by defining new routes.
* Etc.

**Database:**

* The design of the DB must have a balance between performance, volume of information and the 3rd normal form.
* Each table must be analyzed separately and check the level of read and write operations, also check the growth over time.
* Requests must always have filters and limits, so as not to stress the server.
* Avoid queries to the DB within cycles.