

Develop With PHP

Final Project

I. Scenario - The Carnival Booking System

For celebrating the new year, we are holding a **5-days long Carnival**. Required by the sponsor of the carnival, we need to implement a “Carnival Booking System” , which will manage users’ registration, reservation and on-site checking-in of the Carnival. This system will allow people to register a user in it, when the user log in, he/she would be able to reserve a ticket for him(or her)-self. When the user arrives to the party place, he/she would be able to “check-in” offering his/her invitation code and corresponding password.

II. Specification

1. Functionality Structure

The system would have two main parts, the reservation module and the check-in module:

- a) The “Reservation Module”, which will allow user to register at home and submit reservation for a specific day of the carnival;
- b) The “Check-in Module”, which is a web page showed at the reception of the carnival;

2. The “Reservation Module”

2.1. The “Login” Module

The login module should allow people to register a new user in the system, and logging in.

To register, the user needs to provide his/her "email", "username" and "password". The password must be repeated to confirm the input. Therefore the register form should be validated as following rules:

- a) Email: required, must be a valid email address;
- b) Username: required, at least 6 letters long, only letters (upper or lower case) numbers and underscore allowed;
- c) Password: required, at least 6 letters long, any visible character is allowed;
- d) Confirm Password: required, must be exactly identical to the "Password";

To login, user needs use the Username(or Email) and corresponding password.

It's is allow to use Laravel's Breeze system to build this module. For saving time, it's not required to create "Reset password" functions in the project.

2.2. The "Dashboard" Page

When the user is successfully logged in, he/she should be leads to a "Dashboard" page, this is the homepage for a logged-in user.

On this page there should be a reservation list and a "New Reservation" link, and of course the Logout link.

In each item of the reservation list, the reserved day(one of Day1 to Day5) and corresponding invitation code should be displayed. (A user can have more than one reservation, each reservation should be assigned a unique invitation code, see detail rules in the next "New Reservation Page" section). And in

each item, there should be a “Cancel” link displayed for deleting that reservation.

Only reservation in the future can be deleted, if the reservation is “Passed” or “Verified” the word “Passed” or “Verified” should be displayed at the place of “Cancel” link. “Verified” means that invitation code has been “Checked In”.

For a user that has no reservation, a message “No Reservation Yet” should be displayed in the list.

2.3. The “New Reservation” Page

On the New Reservation Page, there should be an available day selection for user to select. If “current day” was in the range of 5 days, only days in future should be displayed (for example, if current is the 3rd day of the carnival, only Day 4 and Day 5 should be displayed).

A “Reserve” button will submit the page, if the validation is passed, a new invitation code would be generated for this reservation, and the user should be redirected to the “Dashboard” page. See below for the reservation rules.

Reservation Rules:

1. One user can have AT MOST 3 reservations in total in the 5 days, but can have ONLY 1 reservation per day;
2. Each reservation should have its own invitation code;
3. The invitation code is a 6-letters long string in lower case, and must be unique in the entire system;
4. In each of these 5 days, we allow 10 reservations at maximum, if there’s no more places, system should give a message to the user.

3. The “Check-in” Module

3.1. The “Check-In” Page

The Check-In page is like a login page, just the user should input the invitation code and his/her password for authentication.

When the check-in form is submitted, system must do following checks:

1. If the submitted code is an existing code in the system?
2. If the submitted password is correct for the user who's having this invitation code?
3. If this code has not been validated yet? (Each code can be validated for only once.)

If all check passed, that reservation should be marked as “Verified”, and a welcome page says “Welcome, \${USERNAME}!” should be displayed. After staying on the welcome page for 5 seconds, the page should be redirected back to the “Check-In” Page automatically.

If a reservation is “verified”, its state should be reflected in that user's dashboard.

4. Others

1. Except the Login and Check-In page, all pages should be protected from non-login access, if a user is accessing these pages directly from url, they should be redirected to the Login.
2. Add following 2 parameters to your config file to simulate the days:
 - CARNIVAL_DAYS: integer, value for how long will the Carnival take;

- `CURRENT_DAY`: integer, value from 0 to `CARNIVAL_DAYS`, where 0 indicates a day before the first day of the Carnival.

This means the length of your Carnival should be configurable, and the selection in “New Reservation” page should be dynamic accordingly.

III. Other Requirements

1. It is FORBIDDEN TO COPY CODE FROM OTHERS, student whom is found to copy code in any method, will be disqualified and marked as CHEATER in the score;
2. The project must be developed using Laravel Framework (V8.x);
3. Use Git to manage your code, and your git history must reflect the history how you build the system step by step (at least 3 commits in time);
4. In the submitted work, a README.md file should be included, which includes 2 parts:
 - a) Part 1: Declare the job done by each of your group member;
 - b) Part 2: Instruction for installing your system on a new machine.
(Consider the new machine has already have the PHP, Composer, MySQL and Git installed.)

IV. Scoring Rules

TOTAL: 20 points

1. Login, Dashboard, New Reservation (8 points)
2. Check-In (5 Points)
3. “Others” in Specification Section (2 points)
4. Code Style (2 points)
5. Git Usage (1 Points)

6. README.txt (2 points)

V. Submission

1. **Deadline:** The work should be sent by email **before 2021-06-11 23:59:59 (Beijing Time)**.
2. **Package your work:** The work should be packaged as a .zip file, and be named as **"PHPD_FinalProject_GROUPID.zip"** (eg.**PHPD_FinalProject_01.zip**).
3. **Email subject format:** The email's subject must be named as **"[PHPD_FinalProject] Group XX"**, in which XX is your group ID.

[End Of Document]